

2019 IEEE 15th International Scientific Conference
on Informatics

INFORMATICS 2019

November 20–22, 2019, Poprad, Slovakia

PROCEEDINGS

Editors

William Steingartner

Štefan Korečko

Anikó Szakál

Organized by

Affiliated branch of the Slovak Society for Applied Cybernetics and Informatics at Department of
Computers and Informatics, FEEI TU of Košice

Faculty of Electrical Engineering and Informatics, Technical University of Košice

Association of Slovak Scientific and Technological Societies

IEEE SMCS Technical Committee on Computational Cybernetics

ISBN 978-1-7281-3178-8

IEEE catalog number CFP19E80-PRT

2019 IEEE 15th International Scientific Conference on Informatics

Copyright ©2019 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Copyright and Reprint Permission:

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limit of U.S. copyright law for private use of patrons those articles in this volume that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For reprint or republication permission, email to IEEE Copyrights Manager at pubs-permissions@ieee.org. All rights reserved. Copyright ©2019 by IEEE.

Other copying, reprint or reproduction requests should be addressed to IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

IEEE Catalog Number CFP19E80-PRT
ISBN 978-1-7281-3178-8

Additional copies of this publication are available from
Curran Associates, Inc.
57 Morehouse Lane
Red Hook, NY 12571 USA
+1 845 758 0400
+1 845 758 2633 (FAX)
email: curran@proceedings.com

PREFACE

It is our pleasure to present the proceedings of the 2019 IEEE 15th International Scientific Conference on Informatics. Its success follows from our improvement efforts to publish with higher standards in the various areas of computer science. As a part of this, our conference became a significant international forum for researchers, engineers and young IT experts to exchange and share their experiences and research results about most aspects of science and social research and discuss new ideas, practical challenges encountered and the solutions adopted.

This conference proceedings collection includes papers covering the research works submitted and accepted for the 15th edition of the International Scientific Conference on Informatics.

The topics of this conference cover theoretical and practical results, along with methods for transferring these research results into real-life domains, by scientists and experts working in computer science and computing-related fields. The rôle of the conference is also to provide an opportunity for young researchers to demonstrate their achievements and to discuss their results at an international scientific forum.

The main topics of the conference were the following:

- Computer Architectures,
- Computer Networks,
- Theoretical Informatics,
- Programming Paradigms,
- Programming Languages,
- Software Engineering,
- Distributed Systems,
- Computer Graphics and Virtual Reality,
- Artificial Intelligence,
- Data and Knowledge Bases,
- Intelligent Information Systems,
- Applied Informatics and Simulation and
- Quality of Software and Services.

This year the conference, co-organized by

- Faculty of Electrical Engineering and Informatics, Technical University of Košice,
- Slovak Society for Applied Cybernetics and Informatics, branch at Department of Computers and Informatics,
- Association of Slovak Scientific and Technological Societies and
- IEEE SMCS Technical Committee on Computational Cybernetics,

is held on November 20th-22nd, 2019, in Poprad, the biggest town of the Spiš region, the main “entrance gate” to the mountain range of the High Tatras and the centre of the rapidly developing region.

Around 133 proposals have been submitted for the conference. All submitted papers have been peer-reviewed by at least two independent external referees and acceptance was based on the quality and relevance of the research. In the end, 83 papers have been accepted and recommended for presentation. This year we also had two outstanding keynote speakers, both of them with high influence in academia and industry.

We would like to thank the organizing staff, the members of the program committee and reviewers. They have worked very hard in reviewing papers and making valuable suggestions for the authors to improve their works. We also would like to express our gratitude to the external reviewers, for providing extra help in the review process, and the authors for contributing their research result to the conference. Finally, we would like to extend our thanks to all the authors and keynote speakers, who contributed and guaranteed the high and professional standard of this conference.

We also thank the sponsors

- IEEE Hungary Section,
- IEEE SMC Chapter, Hungary,
- IEEE Joint IES/RAS Chapter, Hungary

and the technical co-sponsor

- IEEE SMC Society.

It is with deep regret to share that the founding member of this conference and the long-term president of Slovak Society of Applied Cybernetics and Informatics, Professor Ivan Plander, has recently left us forever. We would like to express our most sincere honour to his memory. Professor Mikuláš Alexík has been elected as the new president of the society.

We wish all attendees of the 2019 IEEE 15th International Scientific Conference on Informatics an enjoyable scientific gathering in Poprad, Slovakia. We look forward to seeing all of you at the next event of the conference.

Poprad, November 2019

On behalf of the Programme and Organizing Committees
William Steingartner

COMMITTEES

General Chair

Liberios Vokorokos, *Dean of Faculty of Electrical Engineering and Informatics, Technical University of Košice (SK)*

Honorary Chair

Mikuláš Alexík, *University of Žilina (SK)*

Program Chair

Valerie Novitzká, *Technical University of Košice (SK)*

Program Committee

Miklós Bartha, *Memorial University of Newfoundland (CA)*

Dmitriy B. Buy, *National University of Taras Shevchenko, Kyiv (UA)*

Jan Čapek, *University of Pardubice (CZ)*

Zbigniew Domański, *Częstochowa University of Technology (PL)*

Erik Duval, *Katholieke Universiteit Leuven (BE)*

Dimitar Filev, *Ohio State University (US)*

Zoltán Fülöp, *University of Szeged (HU)*

Gianina Gábor, *University of Oradea (RO)*

Darko Galinec, *Zagreb University of Applied Sciences (HR)*

Ján Genči, *Technical University of Košice (SK)*

Andrzej Grzybowski, *Częstochowa University of Technology (PL)*

Klaus Haenssger, *Leipzig University of Applied Sciences (DE)*

Tamás Haidegger, *Óbuda University, Budapest (HU)*

Aboul Ella Hassanien, *Cairo University, Giza (EG)*

Zdeněk Havlice, *Technical University of Košice (SK)*

Pedro Rangel Henriques, *University of Minho, Braga (PT)*

Pavel Herout, *University of West Bohemia, Pilsen (CZ)*

Ladislav Hluchý, *Slovak Academy of Sciences, Bratislava (SK)*

Elke Hochmüller, *Carinthia University of Applied Sciences (AT)*

László Horváth, *Óbuda University, Budapest (HU)*

Zoltán Horváth, *Lóránd Eötvös University, Budapest (HU)*

Ladislav Huraj, *University of Ss. Cyril and Methodius in Trnava (SK)*

Péter Kádár, *Óbuda University, Budapest (HU)*

Waldemar W. Koczkodaj, *Laurentian University (CA)*

Levente Kovács, *Óbuda University, Budapest (HU)*

Volodymyr Kovalchuk, *Institute of Water Problems and Land Reclamation, Kyiv (UA)*

Sandra Lovrenčić, *University of Zagreb, Varaždin (HR)*

Mária Lucká, *Slovak University of Technology, Bratislava (SK)*

Ivan Luković, *University of Novi Sad (RS)*

Dragan Mašulović, *University of Novi Sad (RS)*

Karol Matiaško, *University of Žilina (SK)*

Marjan Mernik, *University of Maribor (SI)*

Jurij Mihelič, *University of Ljubljana (SI)*

Hanspeter Mössenböck, *Johannes Kepler University Linz (AT)*

Günter Müller, *Albert-Ludwigs-Universität Freiburg (DE)*

Hiroshi Nakano, *Kumamoto University (JP)*

Mykola S. Nikitchenko, *National University of Taras Shevchenko, Kyiv (UA)*

Lucia Pomello, *University of Milano-Bicocca (IT)*

Herbert Prähoffer, *Johannes Kepler University Linz (AT)*

Horia F. Pop, *Babes Bolyai University, Cluj (RO)*

Stanislav Racek, *University of West Bohemia, Pilsen (CZ)*

Mihály Réger, *Óbuda University, Budapest (HU)*

Sonja Ristić, *University of Novi Sad (RS)*

Imre J. Rudas, *Óbuda University, Budapest (HU)*

Gábor Sági, *Hungarian Academy of Sciences, Budapest (HU)*

Abdel-Badeeh M. Salem, *Ain Shams University, Cairo (EG)*
Wolfgang Schreiner, *RISC, Johannes Kepler University Linz (AT)*
Elena Somova, *University of Plovdiv (BG)*
William Steingartner, *Technical University of Košice (SK)*
Jiří Šafařík, *University of West Bohemia, Pilsen (CZ)*
Petr Šaloun, *VŠB – Technical University of Ostrava (CZ)*
Jarmila Škrinárová, *Matej Bel University, Banská Bystrica (SK)*
Michal Štepanovský, *Czech Technical University in Prague (CZ)*
József K. Tar, *Óbuda University, Budapest (HU)*
Katarína Teplická, *Technical University of Košice (SK)*
Renáta Tkáčová, *Technical University of Košice (SK)*
Tsuyoshi Usagawa, *Kumamoto University (JP)*
Valentino Vranić, *Slovak University of Technology, Bratislava (SK)*
Neven Vrček, *University of Zagreb (HR)*
František Zbořil, *Brno University of Technology (CZ)*
Jaroslav Zendulka, *Brno University of Technology (CZ)*
Nataliia Zhukovska, *National University of Water and Environmental Engineering, Rivne (UA)*
Viktor Zhukovskyy, *National University of Water and Environmental Engineering, Rivne (UA)*
Marianna Zichar, *University of Debrecen (HU)*
Doina Zmaranda, *University of Oradea (RO)*

Organizing Committee

Milan Šujanský, *Technical University of Košice (SK)* (chair)
Anikó Szakál, *Óbuda University, Budapest (HU)* (financial chair)
William Steingartner, *Technical University of Košice (SK)* (general manager)
Martina Dragošeková, *Technical University of Košice (SK)*
Sergej Chodarev, *Technical University of Košice (SK)*
Štefan Korečko, *Technical University of Košice (SK)*
Ján Perháč, *Technical University of Košice (SK)*

Technical staff

Marián Hudák, *Technical University of Košice (SK)*
Tomáš Tarkanič, *Technical University of Košice (SK)*

ACKNOWLEDGEMENT TO REVIEWERS

We would like to gratefully appreciate the following distinguished reviewers for spending their invaluable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a positive impact on the quality of the contributions to the 2019 IEEE 15th International Scientific Conference on Informatics:

Mikuláš Alexík	Mirosław Kurkowski	Tetiana Shestakevych
Andraš Anderla	Michal Kvet	Olga Siedlecka-Lamch
Gabriela Andrejková	Dominik Lakatoš	Vladimír Siládi
Norbert Ádám	Lucie Lendelová	Branislav Sobota
František Babič	Attila Lovas	Pavol Sokol
Igor Balaban	Sandra Lovrenčić	Elena Somova
Anton Baláž	Mária Lucká	Michael Sonntag
Igor Bandurič	Marián Mach	Zlatko Stapić
Michaela Bačíková	Kristína Machová	Michal Staš
Peter Bednár	Branislav Madoš	Jozef Suchý
Zuzana Bilanová	Ján Magyar	William Steingartner
Janusz Bobulski	Eva Malichová	Matúš Sulír
Ján Boháčik	Dragan Mašulović	János Sztrik
Ján Buša	Ana Meštrović	Sabina Szymoniak
Luboš Buzna	Miroslav Michalko	Slavomír Šimoňák
Dawid Cekus	Alžbeta Michalíková	Jarmila Škrinárová
David Cerna	Jurij Mihelič	Martin Štancel
Sergej Chodarev	Szilvia Nagy	Michal Štepanovský
Grigoreta Cojocar	Gabriela Nečasová	Jana Šťastná
Tomasz Derda	Mykola Nikiktchenko	Krzysztof Talaška
József Dombi	Valerie Novitzká	Máté Tejfel
Peter Drotár	Dijana Oreški	Katarína Teplická
Adam Dudáš	Luboš Ovseník	Oleg Tikhonenko
Emil Evin	Ján Paralič	Renáta Tkáčová
Lidija Fodor	Piotr Paszta	Emese Tokarčíková
Darko Galinec	Norbert Pataki	Boris Tomaš
Juraj Gazda	Ján Perháč	Katarina Tomičić-Pupek
Zoltán Geler	Dario Pevec	Martin Tomášek
Ján Genči	Igor Pihir	Michal Vagač
Andrzej Grzybowski	Rasislav Pištej	László Varga
Gianina Gábor	Ján Plavka	Ján Vaščák
Dejana Herceg	Zoltán Porkoláb	Petr Veigend
Djordje Herceg	Piotr Puchała	Bálint Vető
Dana Horváthová	Kornelije Rabuzin	Patrik Voštinár
Michal Hulič	Stanislav Racek	Valentino Vranić
Ladislav Huraj	Davorka Radaković	Neven Vrček
Marián Jenčík	Danijel Radošević	Wolfgang Windsteiger
Ondrej Kainz	Sonja Ristić	Elena Zaitseva
Melinda Katona	Krzysztof Rojek	František Vítězslav Zbořil
Dragutin Kermek	Mathias Romańczyk	Jaroslav Zendulka
Valentina Kirinić	Jaroslav Rozman	Natalia Zhukovska
Marián Klešč	Gabor Sagi	Viktor Zhukovskyy
Igor Kollár	Petr Šaloun	Marianna Zichar
Jan Kollár	Martin Sarnovský	Doina Zmaranda
Štefan Korečko	Václav Šátek	Iveta Zolotová
Pawel Kossecki	Miloš Savić	
Volodymyr Kovalchuk	Rafał Scherer	
Mariusz Kubanek	Wolfgang Schreiner	

TABLE OF CONTENTS

Invited Papers

<i>Michał Mazur, Mirosław Kurkowski:</i> Complexity of Security Protocols Verification Tools	1
---	---

<i>Wolfgang Schreiner:</i> Logic and Semantic Technologies for Computer Science Education	7
--	---

Regular Papers

<i>David Andrešič, Petr Šaloun, Bronislava Suchánová:</i> Large Astronomical Time Series Pre-processing and Visualization for Classification using Artificial Neural Networks	13
--	----

<i>Dražen Bajer, Bruno Zorić, Mario Dudjak, Goran Martinović:</i> Evaluation and analysis of bio-inspired optimisation algorithms for feature selection	18
--	----

<i>Branislav Bednár, Zuzana Bilanová:</i> Towards Automated Translating Natural Language Sentences into Intensional Constructions	26
--	----

<i>Bianka Flóra Békefi, Kristóf Szabados, Attila Kovács:</i> A case study on the effects and limitations of refactoring	31
--	----

<i>Peter Berta, Valentino Vranić:</i> Synergy of Organizational Patterns and Aspect-Oriented Programming	37
---	----

<i>Ingmar Besic, Zikrija Avdagic, Kerim Hodzic:</i> Virtual Reality Test Setup for Visual Impairment Studies	43
---	----

<i>Bence Bogdándy, Zsolt Tóth:</i> Analysis of Training Parameters of Feed Forward Neural Networks for WiFi RSSI Modeling	49
--	----

<i>Tibor Brunner, Zoltán Porkoláb:</i> The Role of the Version Control Information in Code Comprehension	54
---	----

<i>Hana Bučková, Jiří Dostál:</i> Teachers' attitudes towards implementing coding at schools	60
---	----

<i>Jasmin Cosic, Christian Schlehner, Drazen Morog:</i> New Challenges in Forensic Analysis in Railway Domain	66
--	----

<i>Peter Čerešník, Branislav Madoš, Anton Baláž, Zuzana Bilanová:</i> SSVDAG*: Efficient Volume Data Representation Using Enhanced Symmetry-Aware Sparse Voxel Directed Acyclic Graph	70
--	----

<i>Uroš Čibej, Jurij Mihelič:</i> Two classes of graphs with symmetries allowing a compact representation	76
--	----

<i>Tomaž Dobravec:</i> Java Virtual Machine Educational Tools	82
--	----

<i>István Donkó, János Márk Szalai-Gindl, Gergő Gombos, Attila Kiss:</i> An implementation of the M-tree index structure for PostgreSQL using GiST	87
<i>Adam Dudáš, Jarmila Škrinárová, Eduard Vesel:</i> Optimization design for parallel coloring of a set of graphs in the High-Performance Computing	93
<i>Jana Dudova, Emil Kršák, Tomáš Kello:</i> Pre-diagnostic of diabetes with the use of statistical machine learning	100
<i>Michal Ďuračík, Emil Kršák, Matej Meško, Ján Ružbarský:</i> Software architecture of Automatic Train Operation	106
<i>Baasanjargal Erdenebat, Tamás Kozsik:</i> Adoption of cloud and containerization technologies in Mongolia	110
<i>Endre Fülöp, Norbert Pataki:</i> Symbolic Execution with Finite State Automata	116
<i>Darko Galinec, William Steingartner, Vinko Zebić:</i> Cyber Rapid Response Team: An Option within Hybrid Threats	121
<i>Elzbieta Gawronska:</i> Modeling and computer simulations with the use of a specific organization of calculations	128
<i>Andrzej Z. Grzybowski, Piotr Puchala:</i> A Class of Multidimensional Oscillating Sequences and Related Young Functionals	134
<i>Andrzej Z. Grzybowski, Tomasz Starczewski:</i> Comparative study of inconsistency indices based on a notion of a triad inconsistency	139
<i>Hristina Gulabovska, Zoltán Porkoláb:</i> Towards More Sophisticated Static Analysis Methods of Python Programs	144
<i>Renát Haluška, Luboš Ovseník, Peter Šul'aj:</i> The Use of Mini Computer in Hybrid FSO/RF System	150
<i>János Hollósi, Áron Ballagi:</i> Training Neural Networks with Computer Generated Images	155
<i>Marián Hudák, Štefan Korečko, Branislav Sobota:</i> Enhancing Team Interaction and Cross-platform Access in Web-based Collaborative Virtual Environments ...	160
<i>Ladislav Huraj, Marek Šimon:</i> Realtime attack environment for DDoS experimentation	165
<i>Tomáš Huszaník, Ján Turán, Luboš Ovseník:</i> On the Carrier Suppressed Optical Modulation Methods for High Capacity DWDM Systems	171
<i>Sergej Chodarev, Ján Halama:</i> Interconnecting YAJCo with Xtext: Experience Report	175
<i>Michal Chovanec, Hynek Bachratý, Katarína Jasenčáková, Katarína Bachratá:</i> Influence of CNN Input Modification for Red Blood Cells Trajectory Prediction in Blood Flow	181

<i>Alexandru Iovanovici, Lilla Pellegrini, Anca-Maria Moscovici, Monica Leba:</i> Network motifs uncovering hidden characteristics of urban public transportation	188
<i>Ján Janech, Marek Tavač, Michal Kvet:</i> Versioned database storage using unitemporal relational database	193
<i>Ondrej Kainz, Eduard Karpiel, Rastislav Petija, Miroslav Michalko, Frantisek Jakob:</i> Detection of non-standard situations in smart water metering	198
<i>Paweł Kossecki:</i> Valuation in the Sector of New Technologies – Taxonomic Measures and Comparison of Markets	204
<i>Peter Krammer, Marcel Kvassay, Ondrej Habala, Ladislav Hluchý:</i> Short-Term Rainfall Estimation by Machine Learning Methods	210
<i>Gabor Kruppai, Attila Kiss:</i> Building Neural Networks as Dataflow Graph	216
<i>Michal Kvet, Karol Matiaško:</i> Efficiency of the relational database tuple access	222
<i>Ján Lang, Radovan Kohút:</i> Mind Map and Business Process Model: Specification support by model transformation	228
<i>Małgorzata Łazarska, Olga Siedlecka-Lamch:</i> Comparative study of relational and graph databases	234
<i>Dániel Lukács, Gergely Pongrácz, Máté Teffel:</i> Performance guarantees for P4 through cost analysis	242
<i>Matej Madeja, Jaroslav Porubän:</i> Accuracy of Unit Under Test Identification Using Latent Semantic Analysis and Latent Dirichlet Allocation .	248
<i>Dusan Marcek:</i> Comparison of Predictive Statistical Learning Accuracy with Computational Intelligence Methods	254
<i>Eva Marková, Tomáš Bajtoš, Pavol Sokol, Terézia Mézešová:</i> Classification of malicious emails	260
<i>Michal Márton, Luboš Ovseník, Ján Turán, Tomáš Huszaník, Jakub Urbanský:</i> Design of Microstrip Patch Antennas Operating on 2.45GHz in HFSS and FEKO	266
<i>Marek Menšík, Marie Duží, Adam Albert, Vojtěch Patschka, Miroslav Pajr:</i> Seeking Relevant Information Sources	271
<i>Jurij Mihelič, Uroš Čibej:</i> A comparison of refinement techniques for the backtracking algorithms for the subgraph isomorphism problem	277
<i>Alžbeta Michalíková, Branislav Pažický:</i> Classification of Tire Tread Images by Using Neural Networks	283
<i>Gabriela Nečasová, Petr Veigend, Václav Šátek:</i> High Order Numerical Integration Method and its Applications – The First 36 Years	288

<i>Marie Nedvědová, Alena Pozdílková, Jaroslav Marek, Michael Bažant:</i> A Simulation Study of Depot with Multiple Temperature Mode: Maximum Flow Estimation	294
<i>Boldizsár Németh, Michal Ďuračik:</i> HaskellCompass: Extending the CodeCompass comprehension framework for Haskell	299
<i>Michal Olenčin, Ján Perháč:</i> Automated configuration of a Linux web server security	304
<i>George Papageorgiou, Anastasia Ioannou, Athanasios Maimaris, Alexander N. Ness:</i> Managing The Implementation of a Smart Pedestrian Network (SPN) System	309
<i>George Papageorgiou, Anastasia Ioannou, Athanasios Maimaris, Alexander N. Ness:</i> A Framework for Evaluating the Benefits of Implementing a Smart Pedestrian Network System	315
<i>Patrik Pekarčík, Tomáš Kekeňák, Pavol Sokol, Terézia Mézešová:</i> Real-time processing of cybersecurity system data for attacker profiling	321
<i>Michal Povinský, Miroslav Melicherčík, Vladimír Siládi:</i> A Chatbot based on Deep Neural Network and Public Cloud Services with TJBOT Interface	327
<i>Alena Pozdílková, Marie Nedvědová, Jaroslav Marek:</i> An Approach to Web Advertising Server Parsing – Apartment Price Analysis in the Czech republic	333
<i>Gabor Sagi, Karrar Al-sabti:</i> Totally bounded metric spaces and similarity detecting algorithms	338
<i>Iman Aamara Salem, Radwa Fathalla, Mohamed Kholeif:</i> A Deep Meta-learning framework for Heart Disease Prediction	343
<i>Jozef Sitarčík, Mária Lucká:</i> epiBAT: Multi-objective bat algorithm for detection of epistatic interactions	351
<i>Vaclav Skala, Martin Cervenka:</i> Novel RBF Approximation Method Based on Geometrical Properties for Signal Processing with a New RBF Function: Experimental Comparison	357
<i>Michal Smolik, Vaclav Skala:</i> Reconstruction of Corrupted Vector Fields using Radial Basis Functions	363
<i>Artur Soboń, Mirosław Kurkowski, Sylwia Stachowiak:</i> Towards Complete SAT-based Cryptanalysis of RC5 Cipher	369
<i>Branislav Sobota, Filip Majcher, Martin Sivý, Marián Hudák:</i> Chaos simulation and audio output	375
<i>William Steingartner, Ján Eged, Davorka Radaković, Valerie Novitzká:</i> Some innovations of teaching the course on Data structures and algorithms	380
<i>William Steingartner, Martin Haratim, Jiří Dostál:</i> Software visualization of natural semantics of imperative languages - a teaching tool	387
<i>Matúš Sulír, Jaroslav Porubän:</i> Designing Voice-Controllable APIs	393

<i>Arnold Szederjesi-Dragomir, Radu D. Gaceanu, Horia F. Pop, Costel Sarbu:</i> A Comparison Study of Similarity Measures in Rough Sets Clustering	399
<i>Martin Švejda, Tomáš Myslivec, Martin Goubej:</i> Coordinated jogging algorithm for general serial robot architectures	405
<i>Judit Tamás:</i> Hierarchical Clustering based on IndoorGML Document	411
<i>Katarína Teplická:</i> Significance of implementation of Management Information Systems in the IPO chain for SMEs	417
<i>Oleg Tikhonenko, Marcin Ziólkowski:</i> Queueing Models of Systems with Non-Homogeneous Customers and their Applications in Computer Science	423
<i>Štefan Toth, Emanuel Zaymus, Michal Ďuračík, Matej Meško, Patrik Hrkút:</i> Automatic restoration of diacritics based on word n-grams for Slovak texts	429
<i>Monika Václavková, Marek Kvet, Peter Sedláček:</i> Graphical Development Environment for Object Programming Teaching Support	435
<i>Michal Vagač, Michal Povinský, Miroslav Melicherčík:</i> Obtaining tire tread model from its real world photo	441
<i>Jakub Vasta, Vaclav Skala, Michal Smolik, Martin Cervenka:</i> Modified Radial Basis Functions Approximation Respecting Data Local Features	445
<i>Marko Vještica, Vladimir Dimitrieski, Milan Pisarić, Slavica Kordić, Sonja Ristić, Ivan Luković:</i> Towards a formal description and automatic execution of production processes	450
<i>Patrik Voštinár, Dana Horváthová, Martin Mitter, Martin Bako:</i> Possibilities of human interaction in the virtual environment	456
<i>Xinyue Wang, Nobutada Fujii, Toshiya Kaihara, Daisuke Kokuryo:</i> A proposal for service design based on user's action history using machine learning	461
<i>Ruriko Watanabe, Nobutada Fujii, Daisuke Kokuryo, Toshiya Kaihara, Yoichi Abe, Ryoko Santo:</i> A study on support method of consulting service using text mining -Application to real problem-	467
<i>Marek Žák, Jaroslav Rozman, František V. Zbořil:</i> Design of Omni-directional Hexapod Robot with Horizontal Coxa Joint	473
Author Index	478

Complexity of Security Protocols Verification Tools

1st Michał Mazur

Cardinal St. Wyszyński University
Institute of Computer Science
Warsaw, Poland
michal.mazur@uksw.edu.pl

2nd Mirosław Kurkowski

Cardinal St. Wyszyński University
Institute of Computer Science
Warsaw, Poland
m.kurkowski@uksw.edu.pl

Abstract—Since the eighties of the XX century, many methods and several automatic tools for security protocols verification have been introduced and successfully used. These tools allowed protocols designers to discover many types of attacks upon security solutions in open computer networks. In this paper, we deal with the formal analysis of the complexity of structures used in such tools. In that sense, we are trying to compare two tools developed and successfully used over many years. Firstly, we examine the complexity of the structures used in the SATMC tool, one of the modules of the well-known AVISPA tool, probably the most famous and the most effective in this area. The second analyzed formalism is the cryptographic module of the VerICS tool, constructed for verification of various kinds of concurrent systems. In both cases, we show basic concepts of verification way, present structures used and discuss based on them the protocol verification process. Finally, we try to estimate and compare the complexity of the structures constructed by these tools during the security protocols' automatic verification process.

Index Terms—Security protocols, verification tools, structures complexity.

I. INTRODUCTION TO SECURITY PROTOCOLS

In the era of rapidly developing digital technologies, it is difficult to imagine unsecured connections. The problem of ensuring the security of information sent through a computer network is one of the most important problems of today's network practice. Control systems without secure communication can quickly become lethal to their users. Increasingly, Internet users are paying attention to a small green padlock in the address bar of a web browser. Its presence indicates a secure connection, for example, using the SSL Protocol. SSL is a well-known and widely recognized protocol that ensures secure connectivity in open computer networks. Not so many Internet users know that the most important part of this protocol is a short data exchange scheme called in the literature the Security Protocol (SP) [13].

For many years SP are a very important and key element of security in open computer networks. In systems that use public key infrastructure, these short concurrent schemes/algorithms are often used during information exchange to, in general, ensure an adequate level of data security over networks and electronic transactions. Their main tasks include, among others, mutual authentication of communicating parties (users, servers) or distribution of a new session key. As we mentioned before they are most often used as necessary components of

large systems, such as widely used communication protocols. Important examples of such systems are Kerberos, SSL/TLS and Zfone. However, in the literature and network's practice, sometimes in SP schemes errors have been found. This justified the important need for developing new, automatic methods for verifying SP' properties in order to find errors that allow undesirable behaviors that can reduce the level of security systems or even discredit it completely [3], [12].

The main goals that should be guaranteed by SP are the following:

- 1) one side or mutual authentication of communicating parties,
- 2) keeping confidentiality of information transmitted,
- 3) keeping integrity of information transmitted,
- 4) new session key distribution.

SP applied in practice can realize one or more of the mentioned objectives. They may also use both types of cryptography: symmetric and asymmetric. This is sometimes described in names of protocols: symmetric, authentication or session key distribution protocols. For the sake of simplicity, in further sections, we will call sometimes all SP simply: protocols [13].

The actions performed by parties during the protocol execution can be divided into internal and external ones. By the external actions we mean actions of mutual transmission of information by the parties. Specification of these actions must contain a source of each information (a sender), a receiver of the information and of course, it's content, indicating respectively, which parts of the information have to be encrypted and how. The internal actions are all the other actions that each party needs to perform on its own during the protocol execution. They include, for example, generating new confidential information, encryption, and decryption of cryptograms, comparing data or performing mathematical operations on locally stored data.

Applying protocols in security systems need to fulfill several requirements:

- 1) each user must be aware of its use and has to agree for applying the protocol,
- 2) each user has to know the protocol's scheme and execute consecutively all the steps of the protocol,

- 3) all steps of the protocol must be precisely defined so that the users could not be confused about the way or the order of their executing.

II. PROTOCOL EXAMPLE

Below we present as an example a well known and well-researched protocol: Needham Schroeder Public Key Authentication Protocol (NSPK). This example will help for better understanding the problems connected with the design, practical using and verification of protocols [13].

The protocol is described in the so-called Common Language - a protocol specification language widely used in the literature. Unfortunately, as can be seen, this language does not fully describe all behaviors taking place during the protocol execution. It only shows the scheme for sending messages and message construction, i.e. external actions of the protocol. The description of internal actions and other conditions for implementing the protocol is usually written in natural language.

In the notation presented below, the symbols I_A and I_B denote the identifiers of the users *Alice* and *Bob*, which want to communicate safely with each other. By $\langle X \rangle_{K_A}$ we mean a ciphertext that contains a message X encrypted under *Alice's* public key. Similarly, by $\langle X \rangle_{K_B}$, we mean a ciphertext encrypted by *Bob's* public key. N_A and N_B denote so-called *nonces* - disposable, pseudo-random numbers generated by *Alice* and *Bob* respectively and used for only one execution of a protocol or one communication session. These numbers, as well as appropriate cryptographic keys, have to remain secret during the protocol execution. In the next parts of this chapter, we call these nonces and secret cipher keys simply: secrets. Additionally the notation $Alice \rightarrow Bob : X$ means that the message X is transmitted from *Alice* to *Bob*. In our considerations, it is assumed that sending information implies its receipt by the receiver.

The scheme of NSPK Protocol is the following:

- $\alpha 1. Alice \rightarrow Bob : \langle N_A, I_A \rangle_{K_B},$
- $\alpha 2. Bob \rightarrow Alice : \langle N_A, N_B \rangle_{K_A},$
- $\alpha 3. Alice \rightarrow Bob : \langle N_B \rangle_{K_B}.$

In the first protocol's step, *Alice* generates its own nonce N_A and sends it with its identifier to *Bob*. This message is encrypted by the *Bob's* public key. Let us recall that due to rules of asymmetric cryptography only *Bob* can decrypt the message and obtain the secret N_A , because only *Bob* has the private key connected with K_B . In the second step, *Bob* generates a nonce N_B , encrypts it with N_A by *Alice's* public key and sends all to *Alice*. Remember that only *Alice* can decrypt this message. *Alice* compares the number received from *B* with the nonce N_A and at this moment *Alice* acknowledges that *Bob* is authenticated. In the last step, *Alice* encrypts N_B with the key K_B and sends it to *Bob*. *Bob* decrypts the ciphertext (only it is able to do that) and compares the number sent by *Alice* with its own N_B . Here *A* is authenticated by *B*.

This protocol was to guarantee mutual authentication of *Alice* and *Bob*. This NSPK's property has been tested and confirmed by various verification methods many times. After

executing the protocol, users should also be sure that they are the only owners of numbers N_A and N_B . They can now use them in the next communication sessions as identifiers or components of a new cryptographic key. The history of using NSPK is very interesting and informative. NSPK had been used in computer networks for over 17 years. However, in 1995 it appeared that the protocol is not correct. Gavin Lowe discovered an attack upon the protocol [11]. The attack can be carried out by an Intruder labeled by *Intruder*. The Intruder is a user of a computer network, which therefore has its own identifier and asymmetric keys. *Intruder* does not have to execute the protocol strictly according to the scheme. He can cheat other users that communicate with them. Below we present the Lowe attack. After its execution, one of the users is cheated about the identity of another one.

- $\alpha 1. Alice \rightarrow Intruder : \langle N_A, I_A \rangle_{K_I},$
- $\beta 1. Intruder(Alice) \rightarrow B : \langle N_A, I_A \rangle_{K_B},$
- $\beta 2. B \rightarrow Intruder(Alice) : \langle N_A, N_B \rangle_{K_A},$
- $\alpha 2. Intruder \rightarrow Alice : \langle N_A, N_B \rangle_{K_A},$
- $\alpha 3. Alice \rightarrow Intruder : \langle N_B \rangle_{K_I},$
- $\beta 3. Intruder(Alice) \rightarrow B : \langle N_B \rangle_{K_B}.$

The scheme presented above shows two dependent and simultaneous runs of the protocol. The execution α describes the communication of *Alice* with the *Intruder*. The execution β presents the execution between *Intruder* that impersonates *Alice* (we denote it by *Intruder(Alice)*). It can be seen that both these executions cannot be executed independently because of a lack of proper users' knowledge. For example, *Intruder* cannot decrypt $\langle N_A, N_B, I_B \rangle_{K_A}$ and get a nonce N_B . But the scheme of the protocol allows a simultaneous run of both executions because *Intruder* can send an obtained from *Alice* in the step $\beta 2$ the message $\langle N_A, N_B \rangle_{K_A}$ to *Bob* in the step $\alpha 2$. After that, *Bob* is sure that it is communicating with *Alice*, while in fact *Bob* communicates with the *Intruder*. In further communication, the *Intruder* can use numbers N_A and N_B , for example, to deceive *Bob*.

Lowe proposed the fixed version of NSPK as follows [11]:

- $\alpha 1. Alice \rightarrow Bob : \langle N_A, I_A \rangle_{K_B},$
- $\alpha 2. Bob \rightarrow Alice : \langle N_A, N_B, I_B \rangle_{K_A},$
- $\alpha 3. Alice \rightarrow Bob : \langle N_B \rangle_{K_B}.$

The modification in the protocol scheme has been done in the second step. Adding the responder's identifier I_B to the message $\langle N_A, N_B, I_B \rangle_{K_A}$ excludes the possibility of its deceptive use by the *Intruder*. This simple modification prevents the protocol from an attack presented above. The NSPK Protocol corrected by Lowe has passed all correctness verification tests known so far.

The presented above example clearly shows that there is an important need for investigating protocols' properties. In this area, many theoretical and practical solutions were proposed. It is necessary to mention here deductive [3], [10], and inductive [5], [14] methods that were introduced in the nineties of the XX century. Such methods allow discovering many examples of attacks upon protocols. In the case of deduction methods, there were proposed several well constructed logical

systems for describing and deduction about users' behaviors and correctness properties of the protocols. However, in this case, verification requires searching in very huge proof trees. For that reason, the most popular and successfully used for verification are methods based on computations that can be done on different kinds of protocols' semantics [1], [6], [8], [15].

III. SAT-BASED MODEL CHECKER

As we mentioned in the Abstract, in this paper we try to analyze and compare the complexity of structures used for automatic SP verification by SATMC [2] (AVISPA [1]) and VerICS [8]. Unfortunately, structures used for verification strongly depends on the investigated protocol. So far we cannot compute structures complexity in the whole case of SP' verification. Here we try to compute the complexity in the case presented above the NSPK Protocol. We hope that these considerations allow us in the future to compute complexity regardless of the example chosen.

The AVISPA Project (Automated Validation of Internet Security Protocols and Applications) was funded by the European Union in the Future and Emerging Technologies program. It started in 2003. It aimed at developing technology for the analysis of security protocols. The AVISPA tool [1] for semi-automated verification of security protocols was made as one of the achievements. It has four modules each based on different methods – On-the-fly Model-Checker (OFMC [4]), SATbased Model-Checker (SATMC [2]), Constraint-Logic-based Attack Searcher (CL-AtSe [16]), Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). We will focus on SATMC which is based on one of the verification methods called model checker. Authors made the assumptions of perfect cryptography, strong typing and Dolev and Yao (DY) intruder model [7].

To give the capability for users to specify security protocols, members of the AVISPA project made High-Level Protocol Specification Language. The main motivations were to design a language that is accessible to human users and provides a high level of abstraction. It allows specifying a variety of security protocols and their requirements. HLPSL file is automatically translated to Intermediate Format (IF), which is an input file for the AVISPA modules.

In SATMC approach, a protocol verification is modeled by a security problem Θ described later in a paper. The question considered here is the following: whether the protocol satisfies its security requirements. This property is formally presented by a propositional Boolean formula Φ_k which represents all the possible execution of the protocol Θ on paths up to depth k [2]. If the formula is satisfied, then the protocol does not satisfy its requirements. Unsatisfiability of the formula means that there is no attack of the length k but not in general case.

Used in SATMC methodology of SP' verification consists of several steps. Firstly, the protocol should be described in the HLPSL language. Secondly, this specification is translated into the IF language. IF generate rewriting rules that described users' behaviors during protocols executions. At this point,

the problem of achieving security requirements by SP is formalized into a so-called Protocol Insecurity Problem (PIP). Next, PIP is reduced into a Planning Problem (PP). Finally, PP is encoded into a propositional Boolean formula via SAT procedure depending on the time step k , which stands for the length of an attack. Next at the end of verification using SATMC the propositional formula is solved by one of the SAT-solvers. If the formula is satisfiable, then there exists an attack upon the protocol.

The verification method proposed in SATMC and briefly described above seems to be rather complicated. Below we show basic concepts and structures used in this type of verification. Here we step by step present proper, most important definitions. Of course, because of a lack of space, we cannot present and discuss all the details. We only present the most important constructions and compute abstract objects used during verification.

As we mentioned above, one of the first stages of verification is translating IF specification into special rewriting rules.

Rewriting systems are widely used methods of replacing subterms of the formula with other terms. General rules of such systems are given as the expression $t \rightarrow t'$, where t, t' are terms and \rightarrow denote replacing t with t' in a formula. In SATMC authors propose some formalism called (multi)-set rewriting system.

Let $\Theta = \langle \Gamma, B \rangle$ be so-called insecurity problem for a some, chosen protocol. Γ is a rewriting system representing executions of the protocol by honest agents with an assumption of the DY Intruder [7]. B is a set of states that represents violations of security requirements. If there exists a reachable state in B then the protocol is not secure.

Definition 3.1: Rewriting system is a tuple $\Gamma = \langle F, I, L, R \rangle$, where:

- F and L are disjoint sets of atomic formulas in a first-order language, called respectively *facts* and *rule labels*,
- I is a finite subset of 2^F representing the initial states,
- R is a set of labelled rewriting rules of the form

$$F_1 \xrightarrow{a} \exists \mathbf{N} : F_2,$$

where $F_1, F_2 \in F$ are finite, $a \in L$ and $\exists \mathbf{N}$ denotes that variable \mathbf{N} ought to be fresh. We present example of rewriting system, in the case of the NSPK protocol, in the next section.

For two rewriting systems $\Gamma_1 = \langle F_1, I_1, L_1, R_1 \rangle$ and $\Gamma_2 = \langle F_2, I_2, L_2, R_2 \rangle$. *Composition of Γ_1 and Γ_2* is a rewriting system $\Gamma_1 \circ \Gamma_2 = \langle F_1 \cup F_2, L_1 \cup L_2, I, R_1 \cup R_2 \rangle$ with $I = \{S_1 \cup S_2 : S_1 \in I_1, S_2 \in I_2\}$.

The special language consisting of facts and rule labels is based on protocol specification written in HLPSL. It is worth to mention that for each protocol a different such language is generated. It depends for example on how many steps or messages are in protocol.

Each state in the system is represented by a set of facts that are true in this state. Rewrite rule r is applicable in a state S if the left-hand side set of the rule r is a subset of the set of facts representing S . State S' is reachable in Γ if

there exists a path $Ir_0S_1r_1S_2\dots r_iS'$ consisting rules and states where r_j are applicable in S_j for $j = 1, \dots, i$ and also r_0 and r_i are applicable in I and S' respectively.

Protocol Insecurity Problem (PIP) $\Xi = \langle \Gamma, B \rangle$ formally is a reachability problem in the sense mentioned above. Unfortunately finding a solution in the general case is a hard, NP-complete problem. Authors [2] propose a technique called graphpan-based encoding for generating propositional formula Φ_k from Ξ .

Before we can introduce a data structure called the planning graph we should present a number of reductions applied in order to simplify a model.

Definition 3.2: *Planning system* is a tuple $\Pi = \langle \mathcal{F}, \mathcal{A}, \mathcal{I}, \mathcal{O} \rangle$, where:

- \mathcal{F} and \mathcal{A} are disjoint sets of variable-free atomic formulae of a sorted first-order language called *fluents* and *actions* respectively,
- \mathcal{I} is a finite subset of $2^{\mathcal{F}}$ representing the initial states,
- \mathcal{O} is a set of operators of the form

$$Pre \xrightarrow{Act} Add ; Del,$$

where $Act \in \mathcal{A}$ and $Pre, Add, Del \in \mathcal{F}$ such that $Add \cap Del = \emptyset$.

If $Pre_i \xrightarrow{a} Add_i ; Del_i$ and $Pre_j \xrightarrow{a} Add_j ; Del_j$, then $i = j$.

Reachability problem $\Theta = \langle \Pi, G \rangle$ in terms of planning is called planning problem (PP) where Ξ is planning system and \mathcal{G} is a set of goals.

As we can see PIP and PP shared many similarities. One of the significant difference is a form of the rules – rules in PP does not contain quantifiers. The middle step between translation PIP to PP is a translation into quantifier-free PIP. After getting q -free PIP the reduction to PP is simple: let $\Xi = \langle \langle F, I, L, R \rangle, B \rangle$ be protocol insecurity problem. Then the planning problem $\Theta = \langle \langle \mathcal{F}, \mathcal{A}, \mathcal{I}, \mathcal{O} \rangle, \mathcal{G} \rangle$ associated with Ξ is obtained as follows:

- $\mathcal{F} = F$,
- $\mathcal{A} = L$,
- $\mathcal{O} = \{ (F_1\sigma \xrightarrow{a\sigma} F_2\sigma ; F_1\sigma \setminus F_2\sigma : (F_1 \xrightarrow{a} F_2) \in R, \sigma \text{ is a ground substitution}) \}$,
- $\mathcal{I} = I$,
- $\mathcal{G} = B$.

Now we can present the planning graph.

Definition 3.3: Let $\Theta = \langle \mathcal{F}, \mathcal{A}, \mathcal{I}, \mathcal{O} \rangle$ be the planning system. k -planning graph $([\Theta])_k = \langle N_f, N_a, \xrightarrow{pre}, \xrightarrow{add}, \xrightarrow{del}, \oplus_f, \oplus_a \rangle$

for $k > 0$ for Θ is a directed acyclic graph, where:

- $N_f = \{N_f^i\}_{i=0}^k$ is a time-indexed family of sets of *fluent nodes*, where $N_f^i \subset \mathcal{F}$. Similarly $N_a^i \in \mathcal{A}$ for $i = 0, \dots, k-1$ is a set of *action nodes*.
- $\xrightarrow{pre}, \xrightarrow{add}$ and \xrightarrow{del} are vertices corresponding to *Pre, Add* and *Del* of each action $a \in \mathcal{A}$.
- $\oplus_f = \{\oplus_f^i : i = 0, 1, \dots, k\}$ and $\oplus_a = \{\oplus_a^i : i = 0, 1, \dots, k-1\}$ are time-indexed (irreflexive and commutative) relations of mutual exclusion (*mutex*) between fluents and actions respectively.

Extend set \mathcal{A} and \mathcal{O} with the *no-operation actions and operators* of the form $a \xrightarrow{nop(a)} a$; for all $a \in \mathcal{F}$.

As we mentioned before, finding the answer to the question about the security of a protocol is reduced to reachability problem which corresponds to the same problem in the planning graph.

Now we will present how the backward search in k -planning graph $([\Theta])_k$ is done. With each fluent layer N_f^i is associated the set of states that may be reached at step i k .

$$MBR([\Theta])_k, i = \{S \subset N_f^i : \forall f_1, f_2 \in S \neg(f_1 \oplus_f^i f_2)\}.$$

But if exists $S \in MBR([\Theta])_k, i \cap B$ then it does not mean that there exists a path from I to S . It can be easily shown [2].

In the next subsection, we will show the construction of a model. For this purpose, we use the previously introduced NSPK protocol.

Here, let us present language used for the description of states and actions in PIP in the case of NSPK Protocol. Facts describe the behavior of honest agents and intruder differently – agents can provide actions with respect to the protocol and intruder has more actions such as composing messages differently and impersonating. More detail characterization can be found in [2]. In the following section capital letters denote variables and small letters constants.

<i>Fact</i>	::=	<i>PrincipalFact</i> <i>IntruderFact</i>
<i>PrincipalFact</i>	::=	state(<i>StepId</i> , <i>Agent</i> , <i>Agent</i> , <i>Msgs</i> , <i>SessId</i>) msg(<i>StepId</i> , <i>Agent</i> , <i>Agent</i> , <i>Msg</i>) secret(<i>Msg</i> , <i>Agent</i>) witness(<i>Agent</i> , <i>Agent</i> , <i>IndSym</i> , <i>Msg</i>) request(<i>Agent</i> , <i>Agent</i> , <i>IndSym</i> , <i>Msg</i>)
<i>IntruderFact</i>	::=	ik(<i>Msg</i>) msg(<i>StepId</i> , <i>Agent</i> , <i>Agent</i> , <i>Msg</i>)
<i>Msg</i>	::=	<i>Agent</i> <i>NumSym</i> <i>Key</i> { <i>Msg</i> } _{key} \langle <i>Msg</i> , <i>Msg</i> \rangle
	.	.
	.	.
<i>SessId</i>	::=	0 1 2 ...

Figure 1. Grammar of facts

Let $\Xi_{NSPK} = \langle \Gamma_{NSPK}, B_{NSPK} \rangle$ be a protocol insecurity problem associated with a problem of security of the NSPK protocol, where $\Gamma_{NSPK} = \langle F, I, L, R \rangle$. Set of facts F is formed according to the grammar shown above.

The initial state representing two concurrent sessions of NSPK protocol represents set

$$I_0 = \text{state}(0, a, a, [a, i, k_a, k_a^{-1}, k_i], 1). \text{state}(0, a, a, [a, b, k_a, k_a^{-1}, k_b], 2) \\ \cdot \text{state}(1, a, b, [b, a, k_b, k_b^{-1}, k_a], 2) \\ \cdot \text{ik}(i). \text{ik}(a). \text{ik}(b). \text{ik}(k_i). \text{ik}(k_i^{-1}). \text{ik}(k_a). \text{ik}(k_b).$$

Notation $f.f'$ denotes sum of sets $\{f\}$ and $\{f'\}$.

For this protocol, there are 15 rewriting rules. Following rule St_1 describes execution of the first step in NSPK protocol.

$$\text{state}(0, A, A, [A, B, K_a, K_a^{-1}, K_b], S) \\ \xrightarrow{\text{step}_1(A, B, K_a, K_b, S)} \\ \exists N_a :$$

$$\text{state}(2, B, A, [N_a, A, B, K_a, K_a^{-1}, K_b], S). \text{msg}(1, A, B, \{\langle A, N_a \rangle_{K_b}\}) \\ \text{.witness}(A, B, n_a, N_a)$$

With a ground substitution

$$\sigma = [A/a, B/b, K_a/k_a, K_b/k_i, S/1]$$

above rule with a label $\text{step}_1(a, i, k_a, k_b, 1)$ can be executed at the state I . It can be easily shown that with a set of constants $\{a, b, i, k_a, k_b, k_i, 1, 2\}$ and set of variables $\{A, B, K_a, K_b, S\}$ there is 162 ground substitutions.

Set B in a given protocol insecurity problem we can characterize as a set of states in which following propositional formula is satisfied

$$\begin{aligned} &(\text{request}(A, B, n_b, N_b) \wedge \text{witness}(B, A, n_b, N_b)) \\ &\quad \vee \\ &(\text{request}(B, A, n_a, N_a) \wedge \text{witness}(A, B, n_a, N_a)). \end{aligned}$$

The protocol insecurity problem Ξ_{NSPK} is used by all four modules of AVISPA tool. Now we consider a planning problem Θ_{NSPK} and planning graph $([\Theta_{NSPK}])_{12}$. The reduction of Ξ_{NSPK} to Θ_{NSPK} proceeds according to the previously presented algorithm. Whereas facts, initial state, and goals are the same, labels and rules become actions and operators, where there is no longer variables. With 15 rules and 162 ground substitutions there is $15 \times 162 + 2430$ operators. The significant amount of there is useless. For example, consider operator obtained from rule St_1 with a substitution $[A/a, B/a, K_a/k_a, K_b/k_a, S/1]$ which has no syntax error.

The construction of planning graph $([\Theta_{NSPK}])_{12}$ is done by induction – first set of nodes N_f^0 contain all facts from initial state I . Secondly, set N_a^0 of actions contain all actions executable in the initial state. Furthermore, N_f^1 is obtained by including all effects of actions in N_a^0 . Finally, mutex relations are added.

In [2], the authors present the path consisting of 12 rules that corresponds to Lowe's attack. From this, we conclude that this is a 12-planning graph. Further consideration of the complexity of this graph seems to be hard for this moment.

Let us summarize the complexity of the above model. PIP defined for NSPK protocol consists of 15 rewriting rules with 162 ground substitutions, which after compiling to PP gives us 2430 states in the system considered.

IV. SP' VERIFICATION WITH VERICS

Here we briefly present method of SP' verification using VerICS [8].

Let us start with a few words about some SP' specification problems. SP can be described intuitively without a formal definition. The Common Language is one of the popular methods, in which are shown external protocol's actions and the scheme of sending messages. Unfortunately, this is not a complete description of the protocol, since it is necessary to add the information about the internal actions such as generating new confidential information or the process of the message encryption/decryption. A formal definition, which provides all the information about the protocol, can be found in the papers [9], [?]. In these works, a protocol is defined as an

abstract object (algorithm) and it differed from its execution. The executions of the protocol are certain substitutions of the abstract protocol in a defined structure which reflects the actual work of the network. Such substitutions are done combinatorically. Note that in such an approach we can also consider executions, that cannot be done alone. Such executions are treated as hypothetical executions that may be made as parts of the interleaves of several executions that are assembled as the attack. In the case of NSPK Protocol, considering two honest users and the *Intruder* there exist 18 different possible such substitutions/executions.

Consider previously discussed NSPK protocol. Below we present two, hypothetical different executions α^1 and α^2 , both consists of three steps. In the first one, the honest users B and A communicate with each other. In the second: the user A communicates with the Intruder ι who impersonates the user B . Such behavior of the Intruder is marked by $\text{intruder}(B)$. Note that, the second execution cannot be done alone, the Intruder has no possibility to decode information from the user A . However, it can be considered as a potential part of the interleave of several different executions, in which the Intruder can gain the necessary knowledge.

$$\begin{aligned} \alpha_1^1 & \quad Bob & \rightarrow Alice & : \langle N_B, I_B \rangle_{K_A}, \\ \alpha_2^1 & \quad Alice & \rightarrow Bob & : \langle N_B, N_A \rangle_{K_B}, \\ \alpha_3^1 & \quad Bob & \rightarrow Alice & : \langle N_A \rangle_{K_A}. \end{aligned}$$

$$\begin{aligned} \alpha_1^2 & \quad Alice & \rightarrow \text{intruder}(B) & : \langle N_A, I_A \rangle_{K_B}, \\ \alpha_2^2 & \quad \text{intruder}(B) & \rightarrow Alice & : \langle N_A, N_B \rangle_{K_A}, \\ \alpha_3^2 & \quad Alice & \rightarrow \text{intruder}(Bob) & : \langle N_B \rangle_{K_B}. \end{aligned}$$

As we have emphasized earlier, Common Language does not guarantee a full description of the relationships and actions occurring in the protocols. A very important element enabling the construction and analysis of interlaces of various executions are actions of information generation, encryption, and decryption. An even more important aspect is knowing or not knowing about a given communication item. Such "lack" strongly affects the limitation of the searched space of possible performances. We need a full formal model showing the dependencies of the order of the actions being performed and showing the acquired knowledge.

The basic notations used in VerICS method are given below. We begin with defining the following sets:

- $\mathbf{P} = \{P_1, P_2, \dots, P_{n_p}\}$ - a set of the honest participants in the network,
- $\mathbf{P}_\iota = \{\iota, \iota(P_1), \iota(P_2), \dots, \iota(P_{n_p})\}$ - a set of the dishonest participants containing the Intruder and the Intruder impersonating the participant P_i for $1 \leq i \leq n_p$,
- $\mathbf{I} = \{i(P_1), \dots, i(P_{n_p}), i(\iota)\}$ - a set of the identifiers of the participants in the network,
- $\mathbf{K} = \bigcup_{s=1, \dots, N_K} \bigcup_{i, j=1, \dots, n_p, i \neq j} \{K_{ij}^s\} \cup \bigcup_{i=1}^{n_p} \{K_{P_i}, K_{P_i}^{-1}\} \cup \{K_\iota, K_\iota^{-1}\}$ - a set of the symmetric and asymmetric cryptographic keys (already existing or possible to be generated) of the participants,

– $\mathbf{N} = \bigcup_{i=1}^{N_P} \{N_{P_i}^1, \dots, N_{P_i}^{k_N}\} \cup \{N_l^1, \dots, N_l^{k_N}\}$ – a set of the nonces.

Definition 4.1: By the set of letters \mathbf{L} we mean the smallest set fulfilling the following conditions:

- 1) $\mathbf{P} \cup \mathbf{P}_i \cup \mathbf{I} \cup \mathbf{K} \cup \mathbf{N} \subseteq \mathbf{L}$,
- 2) If $X, Y \in \mathbf{L}$, then the concatenation $X \cdot Y \in \mathbf{L}$,
- 3) If $X \in \mathbf{L}$ and $K \in \mathbf{K}$, then $\langle X \rangle_K \in \mathbf{L}$, $\langle X \rangle_K$ is a ciphertext consisting of the letter X encrypted with the key K .

Definition 4.2: Let $X \subseteq \mathbf{L}$ and $K \subseteq \mathbf{K}$. Define the set $\kappa_K(X) \subseteq \mathbf{L}$ as the smallest set of letters satisfying the following conditions:

- 1) $X \subseteq \kappa_K(X)$,
- 2) if $l \cdot m \in \kappa_K(X)$, then $l \in \kappa_K(X)$ and $m \in \kappa_K(X)$,
- 3) if $\langle l \rangle_k \in \kappa_K(X)$ and $k \in \kappa_K(X) \cup K$, then $l \in \kappa_K(X)$.

The set $\kappa_K(X)$ contains all the letters which can be retrieved from X by decomposing a concatenation or decrypting a letter using a key, which is either in $\kappa_K(X)$ or in K . By $\kappa(X)$ we mean the set $\kappa_{\emptyset}(X)$.

The same as in the papers [9] we define the protocol as a sequence of steps defined as ordered five-tuples $\alpha = (S, R, M, G, Kn)$. In such step S is the step initiator (sending part), R is a message recipient, M is a sent message, G is a set of information required in order to be generated by S for the execution of the step α and Kn is a set of information required for S in order to send M .

Consider the execution from the example presented above, we have the following notations:

$$\begin{aligned} \alpha_1^1 &= (B, A, \langle N_B \cdot i(B) \rangle_{K_A}, \{N_B\}, \{i(B), N_B, K_A\}) \\ \alpha_2^1 &= (A, B, \langle N_B \cdot N_A \rangle_{K_B}, \{N_A\}, \{N_B, N_A, K_B\}) \\ \alpha_3^1 &= (B, A, \langle N_A \rangle_{K_A}, \emptyset, \{N_A, K_A\}). \end{aligned}$$

Another examples are the following:

$$\begin{aligned} \alpha_1^3 &= (B, A, \langle N_B \cdot i(B) \rangle_{K_A}, \{N_B\}, \{i(B), N_B, K_A\}), \\ \alpha_2^3 &= (B, A, \langle N_A \cdot N_B \rangle_{K_A}, \{N_B\}, \{N_A, N_B, K_A\}), \\ \alpha_3^3 &= (A, B, \langle N_B \rangle_{K_B}, \emptyset, \{N_B, K_B\}). \\ \alpha_1^4 &= (A, C, \langle N_A \cdot i(A) \rangle_{K_C}, \{N_A\}, \{i(A), N_A, K_C\}), \\ \alpha_2^4 &= (C, A, \langle N_A \cdot N_C \rangle_{K_A}, \{N_C\}, \{N_A, N_C, K_A\}), \\ \alpha_3^4 &= (A, C, \langle N_C \rangle_{K_C}, \emptyset, \{N_C, K_C\}). \end{aligned}$$

In this way, we can describe any number of executions of the tested protocol. During the verification, the considered space is limited to two honest users and the Intruder (more information can be obtained in [9]).

Presented above executions are translated into a product of synchronized automata. In this product automaton, we try to find a run that ends in a state that corresponds to an attack. In this case, verification is finished using SAT solvers too, because the reachability of an unexpected state is encoded into a propositional Boolean formula.

According to our considerations devoted to the complexity of structures used for SP' verification, we can note that used by VerICS structures for verification of NSPK protocol consists of 18 execution automata and 17 knowledge automata. Both kinds of all these automata consist of 88 states in total only. That is the whole structure that allows discovering of Lowe's attack using VerICS.

V. CONCLUSION AND FUTURE WORK

As can be seen, the SP' verification problem is not easy. For this several automatic tools have been introduced and successfully used. Considerations presented in this paper show how many methodologies and different structures can be adapted for solving this problem. We have shown that solutions dedicated strictly to an investigated problem and based on well-grounded rules of mathematical first-order logic are suitable and clear.

In our next work, we will try to compare SATMC and VerICS in a more formal way. We will try to strictly compute the complexity of structures used for verification and complexity functions of algorithms that works over these structures. Next, we will try to do comparisons of VerICS structures complexity with structures used by other verification tools: other AVISPA modules, Scyther and ProVerif. We hope that formal computing of the complexity functions can be done in this case.

REFERENCES

- [1] Armando, A., et. al.: The AVISPA tool for the automated validation of internet security protocols and applications. In: Proc. of 17th Int. Conf. on Computer Aided Verification (CAV'05), vol. 3576 of LNCS, 281–285, Springer Verlag (2005)
- [2] Armando, A., Compagna, L.: SAT-based model-checking for security protocols analysis. International Journal of Information Security, 7(1):3–32, (2008)
- [3] Burrows, M., Abadi, M., and Needham, R. M.: A logic of authentication. ACM Trans. Comput. Syst., 8(1):18–36, (1990)
- [4] Basin, D., Modersheim, S., and Vigano, L.: OFMC: A Symbolic Model-Checker for Security Protocols. International Journal of Information Security, vol. 4, pp. 181–208, (2004)
- [5] Bella, G., Paulson L.C.: Using Isabelle to prove properties of the kerberos authentication system. In H. Orman and C. Meadows, editors, Proc. of the DIMACS Workshop on Design and Formal Verification of Security Protocols, (1997)
- [6] Cremers C., Mauw S.: Operational Semantics and Verification of Security Protocols. Information Security and Cryptography series, Springer, (2012)
- [7] Dolev, D. and Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory, 29(2):198–207, (1983)
- [8] Kacprzak, M., et. al.: Verics 2007 - a model checker for knowledge and real-time. Fundam. Inform., 85(1-4), 313–328, (2008)
- [9] Kurkowski, M., Penczek, W.: Verifying Security Protocols Modeled by Networks of Automata, Fund. Inform., Vol. 79 (3-4), 453–471, IOS Press (2007)
- [10] Kurkowski, M., Srebrny, M.: A Quantifier-free First-order Knowledge Logic of Authentication, Fund. Inform., vol. 72, pp. 263–282, IOS Press (2006)
- [11] Lowe, G.: Breaking and Fixing the Needham-Schroeder Public-key Protocol Using fdr., In TACAS, LNCS, Springer, 147–166, (1996)
- [12] Lowe, G.: Some new attacks upon security protocols. In IEEE Computer Society Press, editor, In Proceedings of the Computer Security Foundations Workshop VIII, (1996)
- [13] Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM, 21(12), 993–999, (1978)
- [14] Paulson L.: Inductive Analysis of the Internet Protocol TLS. ACM Transactions on Information and System Security (TISSEC), vol 2 (3), 332–351, (1999)
- [15] Steingartner, W., Novitzká, V., A new approach to semantics of procedures in categorical terms, In Proc. of 13th IEEE Int. Scient. Conf. on Informatics, Poprad, Slovakia, pp: 252–257 (2015)
- [16] Turuani, M.: The CL-Atse Protocol Analyser. In F. Pfenning, editor, Proceedings of 17th International Conference on Rewriting Techniques and Applications, RTA, Lecture Notes in Computer Science, Seattle (WA), Aug. Springer (2006)

Logic and Semantic Technologies for Computer Science Education

Wolfgang Schreiner

Research Institute for Symbolic Computation (RISC)

Johannes Kepler University

Linz, Austria

Wolfgang.Schreiner@risc.jku.at

Abstract—We report on some projects to develop software rooted in formal logic and semantics in order to enhance education in computer science and mathematics. The goal is to let students actively engage with the course material by solving concrete problems where the correctness of a solution is automatically checked; furthermore, if a solution is not correct or the student gets stuck, the software shall provide additional insight and hints that aid the student towards the desired result. In particular, we describe our experience with the RISCAL software, a model checker for mathematical theories and algorithms, in university courses on logic, formal methods, and formal modeling.

Index Terms—logic, semantics, formal verification, model checking, reasoning about programs, computer science education

I. INTRODUCTION

The use of software plays a growing role in education, in particular also in academic courses for computer science and mathematics. Course notes and presentation slides are issued in electronic form and consumed by students on laptops, tablets, and smart phones, sometimes complemented by recorded or even live video presentations. Course management systems are used to organize the workflow of distributing lecturing material, handing out assignments, facilitating discussions, posting and answering questions, submitting and grading results. Courses in mathematics, science, and engineering ask students to use mathematical software systems to elaborate solutions to given problems. Courses on programming ask students for the submission of their programs in electronic form, for manual execution by human teaching assistants, or automatic checking by auto-grader software [1]–[3].

In this paper we discuss the expansion of this toolset by software that has a formal basis rooted in mathematical logic and semantics, where questions respectively problem statements have a precise and machine-understandable meaning and where the correctness of answers respectively problem solutions can be automatically checked; this software is based on technology derived from automated reasoning and program verification such as satisfiability solvers, interactive proving assistants, automated theorem provers, and model checkers.

Supported by the Johannes Kepler University Linz, Linz Institute of Technology (LIT), Project LOGTECHEDU “Logic Technology for Computer Science Education” and by the OEAD WTZ project SK 14/2018 SemTech.

The use of such software gives students the possibility to self-check the correctness of their solutions and use the feedback of the software to correct their errors. The goal is a style of “self-directed learning” where students do not just passively consume educational material but actively interact with it by producing problem solutions with the help of software.

Concretely we are currently pursuing two related projects, “Logic Technologies for Computer Science Education” (LogTechEdu) at Johannes Kepler University (JKU) Linz [4], and “SemTech” at JKU Linz and the Technical University (TU) of Košice [5], that investigate and further develop such tools for their use in computer science education. For instance, William Steingartner and Valerie Novitzká at TU Košice have developed the toolset Jane [6], [7] that illustrates in a graphical way the formal semantics of a simple programming language and that has been applied in various courses.

Our own contribution to this topic is “RISCAL” [8]–[10], a software system for the formalization of theories and the specification and verification of algorithms. While the RISCAL language is based on full first-order logic and is very expressive (allowing, e.g., implicitly defined functions and non-deterministic computations), by its restriction to models of (parametrizable) finite size the validity of formulas is decidable and the correctness of algorithms can be automatically checked [11]. This enables the user to easily validate the adequacy of formalizations respectively find errors in them; this is generally not possible in proving environments [12] or program verifiers [13] where in a sufficiently strong theory the failure to find a proof does not necessarily indicate the invalidity of a formula. The software also provides various means to aid the understanding of results, e.g., by producing counterexamples or by the graphical visualization of evaluation trees [14]. In this paper, we describe our actual experience with the use of RISCAL in academic courses on formal specification and verification, formal modeling, and logic; these courses have addressed various types of audience, computer scientists as well as mathematicians, from absolute beginners to master students in the later phases of their education.

The remainder of this paper is structured as follows: In Section II we sketch the use of the RISCAL software on small examples. In Section III we describe our experience with the use of RISCAL in academic education. Section IV presents our conclusions and outlines further work.

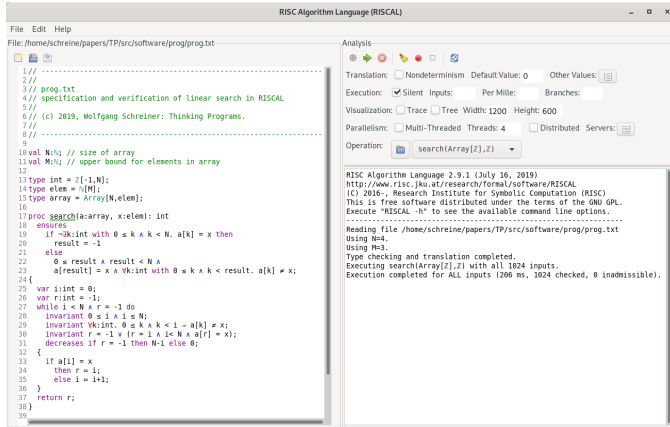


Fig. 1. The RISCAL Graphical User Interface

II. THE RISCAL SOFTWARE

RISCAL (RISC Algorithm Language) is a language and associated software system for formulating theories in first-order logic, describing algorithms in a high-level language, and specifying the behavior of these algorithms by formal contracts. The language is based on a type system where all types have finite sizes (specified by the user); this allows to fully automatically decide formulas and to verify the correctness of algorithms for all possible inputs. To this end, the system translates every syntactic phrase into an executable form of its denotational semantics; the RISCAL model checker evaluates this semantics to determine the results of algorithms and the truth values of formulas such as the postconditions of algorithms. Figure 1 displays the graphical user interface of the software with an editor panel on the left and control elements and an output terminal on the right.

As an example for the use of RISCAL, take the following constant and type declarations:

```
val N:N; val M:N;
type int =  $\mathbb{Z}[-1, N]$ ; type elem =  $\mathbb{N}[M]$ ;
type array = Array[N, elem];
```

These introduce a type `array` of arrays that have some length N and elements of type `elem`; each such element is a natural number up to some maximum M . Likewise an auxiliary type `int` of integers from -1 to N is defined. Furthermore, we introduce by the definition

```
pred occurs(a:array, x:elem, r:int)  $\Leftrightarrow$ 
   $0 \leq r \wedge r < N \wedge a[r] = x;$ 
```

a predicate `occurs` that is true if array a holds element x at position r . From this, the definition

```
fun searchAny(a:array, x:elem): int =
  choose result:int with
  if  $\neg \exists k:int. occurs(a, x, k)$  then
    result = -1
  else
    occurs(a, x, result);
```

defines by an implicit definition a function `searchAny` that returns an arbitrary position at which x occurs in a , respectively the value -1 , if x does not occur in a . RISCAL can execute

this definition, e.g., for $N = 4$ and $M = 3$; if we select for this execution the “nondeterministic” mode, for every input *all* possible outputs are determined:

```
Executing searchAny(Array[Z], Z) with
all 1024 inputs.
Branch 0:0 of nondeterministic function
searchAny([0,0,0,0], 0):
Result (0 ms): 0
Branch 1:0 of nondeterministic function
searchAny([0,0,0,0], 0):
Result (0 ms): 1
...
Branch 4:1023 of nondeterministic function
searchAny([3,3,3,3], 3):
No more results (15 ms).
Execution completed for ALL inputs
(6741 ms, 1024 checked, 0 inadmissible).
```

This is possible, because in nondeterministic mode, the semantics of functions, predicates, and procedures does not denote a single value but a finite set of values which is implemented by a lazily evaluated stream.

Based on predicate `occurs`, we may define and check the (apparently valid) theorem that for every element x there is *some* array that holds x at a unique position:

```
theorem someUnique(x:elem)  $\Leftrightarrow \exists a:array.
  \forall r1:int with occurs(a, x, r1).
  \forall r2:int with occurs(a, x, r2). r1 = r2;$ 
Executing someUnique( $\mathbb{Z}$ ) with all 4 inputs.
Execution completed for ALL inputs (...).
```

However, we may also define and check the (indeed invalid) theorem that above is true for every array:

```
theorem allUnique(x:elem)  $\Leftrightarrow \forall a:array.
  \forall r1:int with occurs(a, x, r1).
  \forall r2:int with occurs(a, x, r2). r1 = r2;$ 
Executing allUnique( $\mathbb{Z}$ ) with all 4 inputs.
ERROR in execution of allUnique(0):
evaluation of
allUnique
at line 33 in file prog.txt:
theorem is not true
ERROR encountered in execution.
```

We may then also ask the system for a counterexample:

```
Executing allUnique_refute().
This sequence of variable assignments
leads to a counterexample ...:
x=0
a=[0,0,0,0]
r1=0
r2=1
Execution completed (5 ms).
```

The truth values of formulas can be also visualized in RISCAL by the help of “pruned evaluation trees” (see Figure 2 for an example of such a tree) that depict those paths in the evaluation of quantified formulas that determine the overall outcome.

After these preliminaries, we define a deterministic procedure that returns the *smallest* position r of element x in array a ; if x does not occur in a , the procedure returns -1 :

```
proc search(a:array, x:elem): int {
  var i:int = 0; var r:int = -1;
  while i < N  $\wedge$  r = -1 do {
    if a[i] = x
```

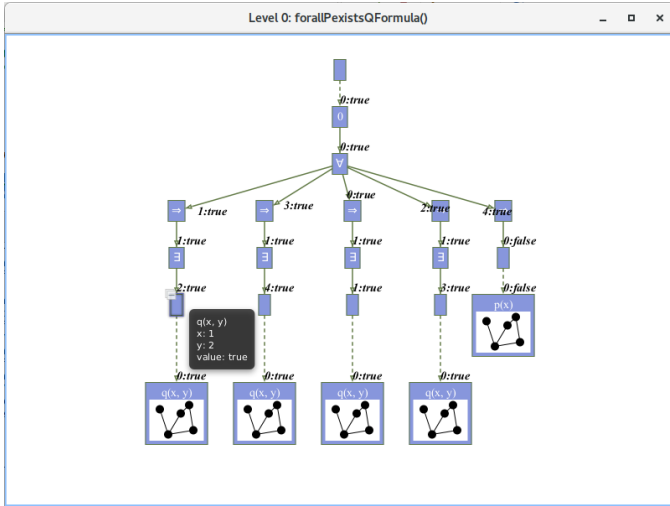


Fig. 2. A Pruned Formula Evaluation Tree in RISCAL

```

    then r := i;
    else i := i+1;
  }
  return r;
}

```

To verify its correctness, we annotate the procedure with the following postcondition:

```

ensures
  if  $\neg \exists k: \text{int. occurs}(a, x, k)$  then
    result = -1
  else  $\text{occurs}(a, x, \text{result}) \wedge$ 
     $\forall k: \text{int with occurs}(a, x, k). \text{result} \leq k;$ 

```

We may then check the correctness of the algorithm for all possible inputs:

```

Executing search(...) with all 1024 inputs.
Execution completed for ALL inputs (...).

```

However, if we replace the test $a[i] = x$ erroneously by $a[i] \neq x$, we get the following error message:

```

ERROR in execution of search([0,0,0,0],0):
evaluation of
  ensures if  $\neg(\exists k: \text{int. occurs}(a, x, k)) \dots$ 
  at line 40 in file prog.txt:
  postcondition is violated by result -1 for
  application search([0,0,0,0], 0)

```

Furthermore, we may annotate the loop in above procedure by an invariant and termination measure:

```

invariant  $0 \leq i \wedge i \leq N;$ 
invariant  $\forall k: \text{int. } 0 \leq k \wedge k < i \Rightarrow a[k] \neq x;$ 
invariant  $r = -1 \vee (r = i \wedge i < N \wedge a[r] = x);$ 
decreases if  $r = -1$  then  $N-i$  else  $0;$ 

```

Every execution of the procedure then checks the validity of these annotations; this in particular ensures that the given invariant is not too strong. However, we may also let the system generate from this invariant verification conditions whose validity implies the correctness of the program. These conditions are theorems that can be automatically checked in RISCAL by a single mouse-click (see Figure 3), which ensures that the specified invariant are strong enough (they

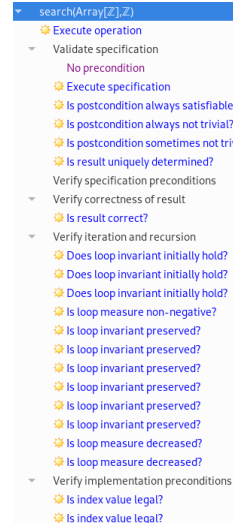


Fig. 3. The Validity of Verification Conditions in RISCAL

are “inductive”). Thus we may subsequently use some other environment to verify the correctness of the algorithm by formal proof for *arbitrary* values of M and N .

The big advantage of RISCAL is that it allows to formulate rich formal/mathematical/logical contents (theories and algorithms) in an expressive language (first order logic including expressions that do not necessarily have unique values) and still have its adequacy fully automatically checked over small domains. In this way, errors in the formulations can be easily caught and thus the formalism be quickly validated; this is not so simply possible with proof-based approaches where failed proof attempts more often than not indicate the inadequacy of proof strategies rather than the invalidity of proof goals. Only when we are after such a validation reasonably convinced about the correctness of the formulations, we will turn to proof-based verification over general domains.

III. LOGIC AND SEMANTIC SOFTWARE FOR EDUCATION

The intent of the projects LogTechEdu and SemTech is to further advance education in computer science and related topics: by utilizing the power of modern software based on formal logic and semantics, students shall engage with the material they encounter by actively producing problem solutions rather than just passively consuming them from the lecturer. For this purpose, we have experimented with various pieces of related software respectively further developed such software. For instance, David Cerna has in the frame of LogTechEdu at JKU Linz recently developed an Android app “AXolotl” for the touch-based training of first-order reasoning with term matching respectively substitution; likewise, William Steingartner and Valerie Novitzká have in the frame of SemTech at TU Košice developed the toolset “Jane” for the semantics-based execution and visualization of a simple procedural programming language that has been successfully employed in various courses on programming language semantics.

However, in this paper we will discuss in more detail those activities that the author has been directly connected to, mostly related to the RISCAL software that was described in the previous section.

a) *Formal Specification and Verification*: Since 2005, the author has given at JKU Linz a yearly 4.5 ECTS course on “Formal Methods in Software Development”; the goal is to educate master students in computer science and computer mathematics in the formal specification and verification of computer programs with the help of various freely available software environments. Since 2009 we have also used our own “RISC ProofNavigator” and since 2011 our “RISC ProgramExplorer” for the purpose of verifying programs by deriving and proving verification conditions. In this course groups are of moderate size (about 25 participants) and have already considerable technical background (their formal background, however, is varying).

However, by relying on proof-based verification tools, the adequacy of formal specifications and annotations (loop invariants) could be only judged by proving the validity of the generated verification conditions. If such proofs did not immediately succeed (after applying some standard interactions with the respective proof assistants), many students were not really able to deduce from the failed proof attempt whether this was due to an inadequate proof strategy or due to deficiencies in the specifications/annotations; moreover, sometimes verification attempts trivially succeeded because preconditions were unsatisfiable or postconditions were generally valid.

In 2017, we introduced RISCAL into the course, replacing some of the initial use of the RISC ProofNavigator/ProgramExplorer. In a first exercise students had to validate specifications with the help of various techniques integrated into RISCAL, such as executing implicitly defined functions that were automatically generated from procedure contracts or checking whether pre- and postconditions satisfied various consistency criteria. In later exercises, students had to annotate procedures with invariants and termination measures, and check the verification conditions that were automatically generated by RISCAL. If conditions were not valid, the RISCAL trace/visualization features could be applied to determine the sources of the errors.

A final anonymous evaluation of the course software performed in 2018 indicated a very high satisfaction of students with RISCAL concerning its ease of use and learning success; the ratings were indeed significantly better than for the proof-based RISC ProofNavigator/ProgramExplorer and summarily higher than for the six other toolsets used in the course (the extended static checker ESC/Java2 being the second most popular one). However, there was no objectively visible effect on exercise grades, which remained mostly in the 80–95% (good to very good) range.

Also outside of JKU Linz, at the Czech Technical University in Prague, Stefan Ratschan used in 2019 RISCAL in a 4.5 ECTS course on “Formal Methods and Specification” for 80 students. The software was applied with very good success; apart from some feedback for improving usability and requests

for additional features, no problems were reported; in 2020, RISCAL will be employed in this course again.

Apart from courses, two bachelor students of computer mathematics used RISCAL to elaborate in their bachelor theses the formal specification and verification of algorithms from discrete mathematics (mostly relating to set and graph theory) respectively for searching and sorting of sequences in various representations (including the major asymptotically fast algorithms). Especially in the later case, it was astonishing to see that the student, without prior expertise in formal verification, was able to come up with sufficiently strong loop invariants to let the verification succeed. These are (anecdotal but nevertheless) strong indications that students that already have a certain formal/technical background are indeed able to develop formally adequate theories and specifications.

b) *Formal Modeling*: In 2019, at JKU a new 3 ECTS course “Formal Modeling” for bachelor students of computer mathematics was introduced, as well as an accompanying proseminar with the same title. The course was given in three modules by three lecturers; in the module “Logic Models of Problems and Computations” of that course (in which about 15 students participated), we applied RISCAL to modeling classical “computational” problems but also “dynamic” dynamic search and scheduling problems, often disguised in the form of “puzzles” such as, e.g., the well-known “goat, wolf, and cabbage” river crossing puzzle.

In contrast to the computational problems specified by a pair of pre- and postconditions, the dynamic problems were modeled in RISCAL by non-deterministic “computational systems” of the following structure:

```

proc system(s0:State):
  Tuple[N[N],Array[N,State]]
  requires init(s0);
{
  var s:State = s0;
  var i:N[N] = 0;
  var t:Array[N,State] = Array[N,Action](s);
  while ¬goal(s) ∧ i < N do
  {
    choose s1:State with next(s,s1);
    s = s1; i = i+1; t[i] = s;
  }
  return (i,t);
}

```

The predicate *init* constrains the initial state of the system; the predicate *goal* describes the desired goal state. Starting with the initial state s_0 as the current state s , the program non-deterministically chooses a successor state s_1 that is related to s by the relation *next*. The computation terminates with the trace t of the the states traversed when a desired goal state has been found or a bound for the number i of steps has been reached (actually, to reduce the search space, typically the computation of the successor state is split into the non-deterministic choice of an “action” a and the subsequent deterministic computation of s_1 from s and a).

Students were handed out specification templates with all the necessary declarations; their task was to formalize the *goal* predicate and the *next* relation. By running the procedure *sys-*

tem in non-deterministic mode, the adequacy of the definitions could be evaluated.

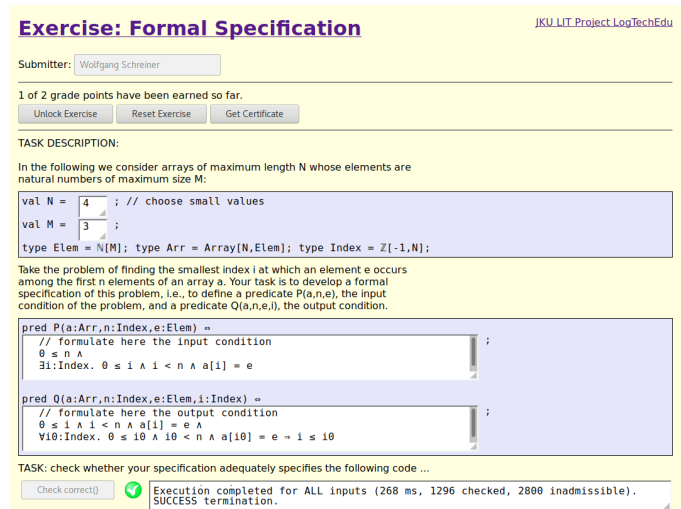
Students apparently liked the “puzzle-like” nature of many of the problems; also, because of the possibility of self-checking, the submitted solutions were indeed mostly correct. In the proseminar (attended by five students), two students choose RISCAL to model self-selected problems, in one case the card game “Uno”, in the other case the problem of the minimization of finite state automata.

All in all, we encountered the use of RISCAL in this novel way a success; it also demonstrated nicely how by the utilization of non-deterministic choices models of “computational systems” can be constructed for which RISCAL was originally not designed. A crucial point, however, here was the appropriate modeling of the system to manage the exponential explosion of the search space of non-deterministic choices.

c) *Logic*: The courses presented so far mainly dealt with moderately sized groups of students with some prior technical and formal knowledge. However, since 2013 we are at JKU also (together with three other lecturers) engaged in a 4.5 ETCS course “Logic” for first semester bachelor students of computer science; this course is attended by 200–250 students most of which have just passed their high school exam, not all with a technical focus. The course is internally organized in three modules “SAT” (propositional logic and satisfiability solving), “FO” (first order predicate logic) and “SMT” (satisfiability modulo theories); the FO module takes half of the course.

Over the years we have also integrated more and more the use of logic-based software tools into the course, a SAT solver (Limboole), an interactive proving assistant (RISC ProofNavigator), an automated prover (Theorema), and SMT solvers (Boolector, Z3). However, prior to 2018, we confined the use of these tools to three optional “laboratory” assignments that students could perform on interest and/or as a substitute for three instances of weekly tests. The main reason was that we could not spend adequate time with the explanation and support of this software and thus did not want to make the use of the software mandatory; consequently, however, only a minority of 5–10% of the students used the software, mainly as a substitute for failed tests. To bring the software more into the “main stream” of the course, we introduced in 2018 weekly “bonus” assignments by which students could earn up to 20% of the grade points for the forthcoming tests. These assignments were of comparatively low complexity; they were mainly intended to raise more interest in the software and thus in the practical aspects of the course.

Also starting with this year, RISCAL replaced the RISC ProofNavigator in the second module FO, specifically in one laboratory assignment and three of the bonus assignments. Questions were handed out in the form of RISCAL skeleton files in which students had to fill in some missing parts; RISCAL itself was provided in the form of a virtual machine (to be downloaded and executed in the free VirtualBox environment) and on a remote server (to be used via an X2go client). In a “syntax” assignment students had to “paren-



Exercise: Formal Specification JKU LIT Project LogTechEdu

Submitter: Wolfgang Schreiner

1 of 2 grade points have been earned so far.

Unlock Exercise Reset Exercise Get Certificate

TASK DESCRIPTION:

In the following we consider arrays of maximum length N whose elements are natural numbers of maximum size M :

```
val N = 4 ; // choose small values
val M = 3 ;
type Elem = N[M]; type Arr = Array(N,Elem); type Index = Z[-1..N];
```

Take the problem of finding the smallest index i at which an element e occurs among the first n elements of an array a . Your task is to develop a formal specification of this problem, i.e., to define a predicate $P(a,n,e)$, the input condition of the problem, and a predicate $Q(a,n,e,i)$, the output condition.

```
pred P(a:Arr,n:Index,e:Elem) =
  // formulate here the input condition
  0 ≤ n ∧
  3i:Index. 0 ≤ i ∧ i < n ∧ a[i] = e ;

pred Q(a:Arr,n:Index,e:Elem,i:Index) =
  // formulate here the output condition
  0 ≤ i ∧ i < n ∧ a[i] = e ∧
  ∀i0:Index. 0 ≤ i0 ∧ i0 < n ∧ a[i0] = e ⇒ i ≤ i0 ;
```

TASK: check whether your specification adequately specifies the following code ...

Check correct! Execution completed for ALL inputs (268 ms, 1296 checked, 2880 inadmissible). SUCCESS termination.

Fig. 4. A RISCAL Web Exercise

thesize” formulas to make their structure unique (RISCAL checked their equivalence to the original unparenthesized formulas) and to translate informal statements into formal ones (RISCAL checked the equivalence to another formalization). In a “semantics” assignment, students had to determine satisfying assignments of first-order formulas (RISCAL checked the correctness of the answers) and to transform formulas into logically equivalent forms with certain syntactic constraints (RISCAL checked here the equivalence). In a “pragmatics” assignment, students had to translate given informal problem specifications (pre- and post-conditions) into logic formulas, which partially involved the definition of auxiliary functions and predicates; RISCAL was here used to validate the results by, e.g., checking the input/output behavior of a function implicitly defined by this condition.

At the end of the course, an anonymous evaluation gave the following results: about 40% of the students performed at least one RISCAL exercise, about the same number reported the use of the software as helpful. These numbers clearly trail the SAT solver Limboole used in the SAT module (about 60% used this one and reported it as helpful) but are also much ahead of the other tools in FO and SMT (used by about 25%). Of those who submitted bonus assignments, most indeed earned the full amount of potential grade points. As for the more general questions on why students used the software, twice as many reported as the main reason to earn the bonus points rather than because of intrinsic interest. Still most positive impact on interest in the course was reported to the software, while most impact on understanding was attributed to the exercises (three times more than to software). While the overall level of grades did not significantly differ from the previous years, we found a strong correlation between performance on bonus assignments and performance in classroom assignments; indeed most students that failed the course did not perform the bonus assignments.

Thus in a nutshell, many students performed the software-

based bonus exercises and those who did so achieved also significantly better results in the classroom exercises. However, weak students (subsequently failing the course) mostly did not use the software. The reason to use the software was mainly the “extrinsic” motivation to earn additional grade points and not an “intrinsic” interest. Nevertheless software was cited as a factor to improve interest in the course, but much less as a factor in improving understanding.

In case of RISCAL, a main deterring factor to use the software may have been the need for a local installation (even if by a virtual machine), the need to learn to use the software, and the need to manipulate text files. In the next 2019 instance of the course, we will therefore use a web-based frontend to a server installation of RISCAL that allows students to perform the exercises within their web browsers (see Figure 4). We hope that this will considerably simplify the use of the software, lower the threshold to use it, and hopefully also allow weaker students to profit from its potential.

d) Correct Program and Algorithm Development: There is one kind of courses where the potential of RISCAL has not yet been applied: those on the development of programs (respectively algorithms) where their correctness with respect to given specifications should be checked. Here lecturers might hand out program assignments in the usual way by the desired interface of a procedure and an informal explanation of the inputs it can expect and the results it must deliver. However, additionally, the procedure would be also equipped with a formal contract, against which the student could fully automatically check the correctness of her solution and the lecturer could fully automatically check the correctness of a submission. If a solution fails the check, the reported error also demonstrates a concrete input/output pair that demonstrates the failure. In this way, in particular, erroneous boundary cases (that very often give problems) can be quickly detected.

We are not in charge of a corresponding course where RISCAL can be applied in this sense but plan in the near future to approach other lecturers teaching program/algorithm development that may find interest in the use of RISCAL.

IV. CONCLUSIONS

From the presented experience, we have good evidence that by the use of software based on formal logic and semantics, such as RISCAL, the education in various areas of science and engineering may be substantially improved. This of course mainly applies to closely related “formal” topics but may be also relevant for topics like programming where RISCAL can check in small domains the correctness of programs with respect to their specifications.

Our main success lies so far on levels of education where students have already at least some prior (technical and/or formal) background. In courses targeted to absolute beginners, it is mainly the stronger students that profit from the software, while the weaker students (already struggling with the basic material) are potentially overwhelmed by the additional “burden” to use software. Here the use of software requires careful evaluation and fine-tuning.

RISCAL itself has reached a stable state and has been applied in various courses, mostly but not only at JKU. Further work will focus on the improvement of feedback mechanisms to help students understand the computed results and the integration with proof-based environments to let RISCAL be used as a “pre-checker” of formalizations in small domains before turning to general proofs in domains of arbitrary size.

REFERENCES

- [1] D. Thiébaud, “Automatic Evaluation of Computer Programs Using Moodle’s Virtual Programming Lab (VPL) Plug-in,” *Journal of Computing Sciences in Colleges*, vol. 30, no. 6, pp. 145–151, Jun. 2015. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2753024.2753053>
- [2] S. V. Yulianto and I. Liem, “Automatic Grader for Programming Assignment using Source Code Analyzer,” in *International Conference on Data and Software Engineering (ICODSE)*. Bandung, Indonesia, November 26–27: IEEE, 2014, pp. 1–4, doi 10.1109/ICODSE.2014.7062687.
- [3] S. H. Edwards and M. A. Perez-Quinones, “Web-CAT: Automatically Grading Programming Assignments,” in *13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE ’08)*. Madrid, Spain, June 30 – July 2: ACM, 2008, p. 328, doi 10.1145/1384271.1384371, see also <http://web-cat.org>.
- [4] LogTechEdu, “JKU LIT Project LOGTECHEDU,” July 2019. [Online]. Available: <http://fmv.jku.at/logtechedu/>
- [5] SemTech, “SemTech — Semantic Technologies for Computer Science Education,” January 2018. [Online]. Available: <https://www.risc.jku.at/projects/SemTech/>
- [6] W. Steingartner and V. Novitzká, “Learning tools in course on semantics of programming languages,” in *MMFT 2017 — Mathematical Modelling in Physics and Engineering*, Czestochowa, Poland, September 18–21, 2017, pp. 137–142. [Online]. Available: http://im.pcz.pl/konferencja/get.php?doc=MMFT2017_streszczenia_wykladow.pdf
- [7] W. Steingartner, M. A. M. Eldojali, D. Radakovic, and J. Dostál, “Software support for course in Semantics of programming languages,” in *IEEE 14th International Scientific Conference on Informatics*, Poprad, Slovakia, November 14–16, 2017, pp. 359–364, doi 10.1109/INFORMATICS.2017.8327275.
- [8] RISCAL, “The RISC Algorithm Language (RISCAL),” March 2017. [Online]. Available: <https://www.risc.jku.at/research/formal/software/RISCAL>
- [9] W. Schreiner, “The RISC Algorithm Language (RISCAL) — Tutorial and Reference Manual (Version 1.0),” RISC, Johannes Kepler University, Linz, Austria, Technical Report, March 2017, available at [8].
- [10] W. Schreiner, A. Brunhumer, and C. Fürst, “Teaching the Formalization of Mathematical Theories and Algorithms via the Automatic Checking of Finite Models,” in *Post-Proceedings ThEdu’17, Theorem proving components for Educational software, Gothenburg, Sweden, August 6, 2017*, ser. EPTCS, P. Quaresma and W. Neuper, Eds., vol. 267, 2018, pp. 120–139, doi 10.4204/EPTCS.267.8.
- [11] W. Schreiner, “Validating Mathematical Theories and Algorithms with RISCAL,” in *CICM 2018, 11th Conference on Intelligent Computer Mathematics, Hagenberg, Austria, August 13–17*, ser. Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence, F. Rabe, W. Farmer, G. Passmore, and A. Youssef, Eds., vol. 11006. Springer, Berlin, 2018, pp. 248–254, doi 10.1007/978-3-319-96812-4_21.
- [12] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Berlin, Germany: Springer, October 2017. [Online]. Available: <http://isabelle.in.tum.de/doc/tutorial.pdf>
- [13] K. R. M. Leino, “Dafny: An Automatic Program Verifier for Functional Correctness,” in *Logic Programming and Automated Reasoning (LPAR-16), Dakar, Senegal, April 25–May 1, 2010*, ser. Lecture Notes in Computer Science, E. M. Clarke and A. Voronkov, Eds., vol. 6355. Springer, Berlin, Germany, 2010, pp. 348–370, doi 10.1007/978-3-642-17511-4_20.
- [14] W. Schreiner, “Theorem and Algorithm Checking for Courses on Logic and Formal Methods,” in *Post-Proceedings ThEdu’18, Theorem proving components for Educational software, Oxford, United Kingdom, July 18, 2018*, ser. EPTCS, P. Quaresma and W. Neuper, Eds., vol. 290, 2019, pp. 56–75, doi 10.4204/EPTCS.290.5.

Large Astronomical Time Series Pre-processing and Visualization for Classification using Artificial Neural Networks

David Andrešič

*Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava, 17. listopadu 15
Ostrava-Poruba, Czech Republic
david.andresic@vsb.cz*

Petr Šaloun

*Faculty of Electrical Engineering and Computer Science
VŠB - Technical University of Ostrava, 17. listopadu 15
Ostrava-Poruba, Czech Republic
petr.saloun@vsb.cz*

Bronislava Suchánová

*Faculty of Informatics and Information Technologies
Slovak university of technology in Bratislava, Ilkovičova 2
Bratislava, Slovakia
xsuchanovab@stuba.sk*

Abstract—Time series analysis is a growing issue in multiple fields of science. One of the most common task is looking for hidden periods in time series data sets. In this work, we have chosen two large astronomical time series collections from BRITE and Kepler K2 projects and analyzed possible approaches for hidden periods search and their classification. Since these data sets are generally large, we were looking for some automated solution based on artificial neural networks that requires some data pre-processing. This work therefore brings a brief overview of possible solutions for looking for hidden periods in astronomical time series with use of pure artificial neural networks or together with more conventional statistical approaches - mainly from a data pre-processing and its visualization point of view.

Index Terms—artificial neural networks, classification, time series, big data

I. INTRODUCTION

Time series are in general discrete sequences of quantity values measured in time. In astronomy, it is usually flux or magnitude on one axis and Julian date on the other axis. These time series comes from multiple observational devices and observatories designed for e.g. variable stars detection, stellar system analysis or extra-solar planets discoveries. Finding periods in these time series is therefore crucial for further research in this area. Since we are talking about large amount of data growing exponentially during last years, we are facing a big data issues, where fast handling and analysis of these amount of data is mandatory.

II. STATE OF THE ART

A conventional way to discover hidden periods in time series data is to use statistical approaches, such as analysis of variance or Lomb-Scargle algorithm. These algorithms attempts

Identify applicable funding agency here. If none, delete this.

to estimate (based on correlations in available data) how probable is some period from the given range of periods. Most literature on time series classification assumes following [1]: copious amounts of perfectly aligned atomic patterns can be obtained, the patterns are all of equal length and every item that we attempt to classify belongs to exactly one of our well-defined classes. Based on these assumptions, the best machine learning approach for time series classification is usually Nearest Neighbour algorithm with the relatively expensive Dynamic Time Warping as the distance measure [2]. But all these assumptions are challenging in astronomical time series since they greatly vary in lengths, periods, noisiness and are without clear borders. In [3] authors also concludes that LSTM is another state-of-the-art technique for this task.

A. Time Series Classification Algorithms

Today, different kinds of (deep) neural networks are favourite to use for time series analysis [3] as briefly mentioned in this section.

1) *Multi-layer Perceptron*: A traditional ANN for classification purposes. ANNs described further in this section can be considered as enhancements of this model. Also, in VI we briefly mention our future intention to improve its classification capabilities by establishing a custom activation function of each artificial neuron using evolutionary techniques.

2) *Long Short-term Memory*: The major drawback of conventional ANNs such as MLP is the inability to react based on previous inputs. This led to a development of Recurrent Neural Network (RNN) [5] that introduces to MLP an internal memory with previous state of hidden layer. LSTM [6] deals with a major drawback of RNN: vanishing and exploding gradients [7] [8]. It seems that noisy time series are quite a challenge for LSTM and RNN in general. Possible solution

to this can be inspired by [9], where the authors used a conversion into a symbolic representation with a self-organizing map.

3) *Convolutional Neural Network*: Traditional pattern recognition usually requires to implement and fine-tune some data pre-processing [10]. The advantage of CNN is that it requires only minimal modification of the original data before classification and its typical usage includes time series classification [10].

4) *Fully-convolutional Neural Network*: Fully-convolutional neural networks (FCN) performs well in fields of time series classification [3]. This kind of network does not require either massive data pre-processing or "feature engineering". Even more, it can itself be used as a tool for significant attributes extraction [3].

5) *LSTM Fully-convolutional Neural Network*: Authors in [3] comes with a new type of ANN called LSTM Fully-convolutional Neural Network (LSTM-FCNN). They suggest to extend the FCN by LSTM module. Authors states that their solution significantly improves the performance of FCN with just a small increase of model size. They also states that their ANN requires minimal data pre-processing and conclude that LSTM-FCNN is superior to current models.

6) *Other Artificial Neural Networks*: We can also consider **Multi-scale Convolutional Neural Networks** (MCNN) which performance is (in general) heavily dependent on time series data pre-processing and optimization of huge amount of hyperparameters. Another option could be **Residual Network** that does not require any either deep data pre-processing or "feature engineering" [3].

B. Known Methods for Time Series Data Pre-processing

Time series, images and even records of human speech usually consists of huge amount of data. If we attempt to classify them using a traditional backpropagation algorithm without attribute extraction, we would face overfitting [10]. In this work, we focus on **significant attributes extraction** that is supposed to reduce the dimensionality of the original data by extracting only significant attributes. These can be various types of statistical attributes or even more complex types such as periodograms or generated images.

According to [3], other often used methods are **Fully-convolutional Neural Network** as a method for significant attribute extraction. In [4], authors also mention **Dynamic time series wraping** and **Shapelet transformation** as a way of time series pre-processing.

III. DATA SETS

We tested our approaches on two different data sets containing thousands of real astronomical time series.

A. BRITE

Data from BRITE project - a group of nanosatellites on lower orbit launched in 2013 and 2014. This data set is made of tabular ASCII files containing (among others) Heliocentric Julian Date and Flux (ADU/s). There are 1119 light

curves within this data set. According to GVCS catalogue [11], 601 of them are of variable stars, 279 not variable and 239 cannot be identified by cross-matching in GCVS catalogue. Beside the fact that **variability data comes from a 3rd party archive**, the greatest disadvantage of this data set is its **variability in sampling rate** (in some cases, intervals between samples are less than 1ms on one side and approximately 60 days on the other side). Another disadvantage is that **number of samples in each light curve varies** (some light curves have less than 10 measurements, while others have tens of thousands of measurements). The average length of light curve is 12642 of samples.

B. Kepler K2

NASA's mission to search Earth-like extra-solar planets in our Galaxy. Data from K2 mission that are subject of this work are publically available [12] [13] and well documented [14] [15]. In this work we are interested in Kepler K2 light curves containing flux of individual objects in time. For these data, Kepler K2 also provides official catalogue of confirmed variable objects that we can utilize.

Based on sampling frequency, we distinguish **2 cadency groups**: *long* with 1765.5s (29.4min) and *short* with 58.89s. On **each Thursday**, more than **160000 objects** with long cadency and **512 objects** with short cadency were measured and archived. The minimal length of measurement was **1/4 of year** for long cadency and **1 month** for short cadency (with exception of Q4 where module 3 objects were lost due to hardware failure). Light curve file is in a form of time series where all undefined values are represented as NaN. As a result, we can obtain about 40000 light curves with length up to 1300 measurements.

IV. PROBLEM SOLUTION

We have decided to compare several methods of time series classification that include use of artificial neural networks. Our primary goal is to find the best method that will classify time series (light curves) at least binary in a sense of object variability: the light curve contains some period (and is therefore of variable object) or whether there is no period found (an object is not variable):

- multi-layer perceptron classification with own activation function,
- recurrent neural network of type LSTM,
- multi-layer perceptron with own activation function in combination with time series pre-processing using Fourier transformation,
- recurrent neural network of type LSTM in combination with time series pre-processing using Fourier transformation,
- multi-layer perceptron classification with own activation function in combination with some method of significant attribute extraction,
- recurrent neural network of type LSTM in combination with some method of significant attribute extraction (a.k.a. feature extraction),

- convolutional neural network with sigmoid activation function,
- other, not so succesfull (in terms of our results) artificial neural networks such as fully-convolutional neural networks, MCDCNN and ResNet.

A. Applied Data Pre-processing Methods

We pre-processed both BRITE and Kepler K2 data sets and transformed the raw data (light curves with flux) into a common form diagestable by the ANN:

- Balancing data sets so it contains same number of variables and not variables.
- Cutting light curves in order to equal their length.
- Mix the data.
- Generate periodogram.
- Perform Fourier transformation.
- Significant attributes extraction in order to reduce dimensionality of time series data (different techniques).
- Data normalization into interval relevant to selected activation function.
- Splitting the data set to training and test set.

1) *Significant Attributes Extraction and Visualization*: During our experiments described in V we discovered that both original data sets may not provide clear examples of time series of variable and non-variable stars. This led to a poor accuracy and we were therefore looking for a way how to distinguish these time series by means of significant attributes extraction. We tested the usability of extracted attributes by visualization using Sammon mapping [16]. Sammon mapping attempts to find a low-dimensionality representation of objects in high-dimensional space with as much respect to their original geometric distances as possible. We used it to convert extracted significant attributes to 2D and visualize, hoping to see clear clusters with variable and non-variable time series. Such set of attributes could be then used for further classification using ANN.

V. EXPERIMENTS AND RESULTS

Not all experiments cover both BRITE and Kepler K2 data sets. We were looking at results with **10 following activation functions**: exponential, sigmoid, hyperbolic tangens, relu, elu, selu, soft plus, softsign, harp sigmoid, linear. Experiments were performed with artificial neural network containing **3 hidden layers: 16 neurons in input, 34 neurons in first hidden, 16 in second hidden and 64 in third hidden layer**.

A. BRITE Data Set

We had started with a more problematic BRITE data set. We attempted to classify time series by several ways, but eventually with poor results.

1) *Multi-layer Perceptron*: The sigmoid activation function was used in the output layer. Training was stopped after 3000 epochs. The learning rate was set to 0.005. For each activation function we trained the MLP 10-times and based on validation data we selected the best model. Its accuracy was then tested on testing data. Results can be seen at Tab. I.

Act. function	Precision	Recall	Accuracy
Exponential	0	null	0.64
Sigmoid	0	null	0.64
Hyperb. tan.	0.03	0.18	0.59
Relu	0	0.60	0.64
Elu	0	null	0.64
Selu	0	null	0.64
Soft plus	0	null	0.64
Soft sign	0	0	0.62
Harp sigmoid	0	null	0.64
Linear	0	0	0.61

TABLE I: Results of MLP classification for BRITE data set.

Act. function	Precision	Recall	Accuracy
Sigmoid	0.98	0.53	0.56
Hyperbolic tangent	0.72	0.61	0.64
Relu	0.67	0.60	0.61
Elu	0.64	0.63	0.63
Selu	0.78	0.62	0.66
Soft plus	0.82	0.58	0.62
Soft sign	0.64	0.58	0.60
Harp sigmoid	0.98	0.49	0.48
Linear	0.73	0.61	0.63

TABLE II: Results of MLP classification for Kepler K2 data set.

Bad results are probably caused by a small data set. Only 538 light curves came out from pre-processing, these were then divided to a training and test set in 70:30 ratio, 10% of training set was used for validation. For the training, only 338 light curves remained. Another issue was the irregular interval between individual measurements within the time series. Based on these results and results with LSTM, we decided to continue only with Kepler K2 data set.

2) *Long Short-term Memory*: In this case, the process was a bit different. We used 900 raw time series (as LSTM is supposed to handle it) cross-matched with GCVS catalogue. We divided them into training and test set in 70:30 ratio. Each time series had up to 66500 measurements ("feature vector"). Shorter time series were padded with -1 to this length and all data normalized. With these settings, we achieved the accuracy of 60%.

B. Kepler K2 Data Set

After attempts with BRITE data set, we switched to a more promising Kepler K2 data set. Configuration was same as in case of BRITE.

1) *Multi-layer Perceptron*: The results with the same configuration as in case of BRITE can be seen in Tab. II. Unfortunately, there is just minimal improvement. The best activation function turned out to be Selu that achieved 0.66 accuracy. Recall is also interesting metric because it is not such an issue if some non-variable object is classified as variable but it is important to minimize the number of undetected variables. From this point of view, the Elu function performed best. Accuracy and loss function of best models is depicted on Fig. 1 and 2.

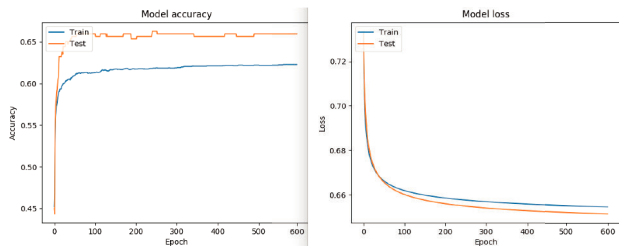


Fig. 1: Results of MLP training with Elu activation function on K2 data set.

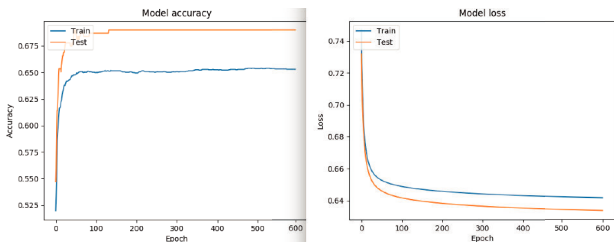


Fig. 2: Results of MLP training with Selu activation function on K2 data set.

Then we attempted to improve the accuracy by generating so-called **periodograms** created by conventional statistical analysis. The classifier then attempted to classify these periodograms instead of light curves. Results can be seen in Tab. III and shows no significant improvement of accuracy. Nevertheless, hyperbolic tangens performed best. We also attempted to establish some **custom activation functions** listed in Tab. IV (raw light curves were used). As last experiment with MLP, we attempted to use our own activation functions with Kepler K2 light curves processed by **Fourier transformation**. Results are listed in Tab. V. We achieved similar results with **Cosine transformation**.

2) *Convolutional Network*: We tried to experiment with CNN in order to compare with MLP. All data from K2 data set were normalized by min-max normalization, we run 5000 epochs with learning rate 0.005 and following network configuration: two convolutional layers with sigmoid act. function, window size of 7 and 6 (or 12) filters, each followed by pooling layer and with output layer with sigmoid activation function. Results are in Tab. VI and on Fig. 3a, 3b, 4.

Act. function	Precision	Recall	Accuracy
Sigmoid	1.00	0.49	0.49
Hyperbolic tangent	0.51	0.71	0.66
Relu	0.54	0.70	0.65
Elu	0.63	0.65	0.65
Selu	0.65	0.61	0.62
Soft plus	0.80	0.61	0.65
Soft sign	0.56	0.67	0.64
Harp sigmoid	0.00	0.00	0.50
Linear	0.61	0.65	0.64

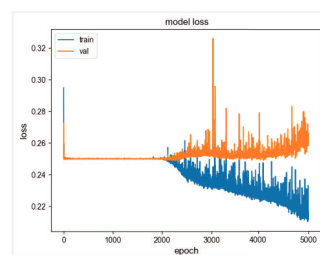
TABLE III: MLP classification for K2 data converted to periodograms.

Act. function	Precision	Recall	Accuracy
$\tanh(0.1*x)$	0	0	0.45
$\tanh(0.3*x)$	0.05	0.53	0.45
$\tanh(0.5*x)$	0.52	0.66	0.59
$\tanh(x)$	0.72	0.68	0.66
$\tanh(1.5*x)$	0.68	0.68	0.65
$\tanh(2*x)$	0.71	0.67	0.65

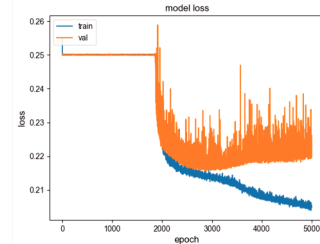
TABLE IV: MLP classification for K2 data using own act. functions.

Act. function	Precision	Recall	Accuracy
$\tanh(1.1*x)$	0.68	0.64	0.64
$\tanh(1.4*x)$	0.75	0.65	0.67
$\tanh(1.5*x)$	0.74	0.65	0.66
$\tanh(1.7*x)$	0.68	0.61	0.61
$\tanh(2*x)$	0.72	0.63	0.63

TABLE V: MLP classification for K2, own act. functions, FT.



(a) CNN experiment #1.



(b) CNN experiment #2.

Fig. 3: CNN experiment loss function chart during training phase.

3) *Other Artificial Neural Networks*: We have been experimenting with several other ANNs including ResNet, Fully-convolutional network, MCD CNN and other configurations of MLP and CNN with even less success than was described above. Their results are therefore omitted from this paper.

4) *Other Significant Attributes Extraction Methods*: As mentioned in IV-A1, we attempted to extract most significant attributes from K2 data set in order to distinguish variable

Light curves	Length	Precision	Acc.	Recall
500	800	0.65	0.65	0.65
7502	1300	0.65	0.64	0.64
500	400	0.64	0.62	0.63

TABLE VI: Results of CNN classification for Kepler K2 data with different count of light curves and measurements in each time series (cutted to this length).

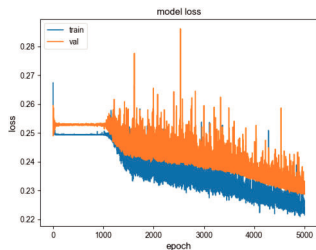


Fig. 4: CNN experiment #3: loss function during training.

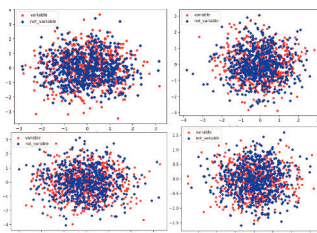


Fig. 5: Sammon projection, random init.: 500 epochs, different time series length (1200-1225 measurements), extracted attributes or just FT or min-max norm.

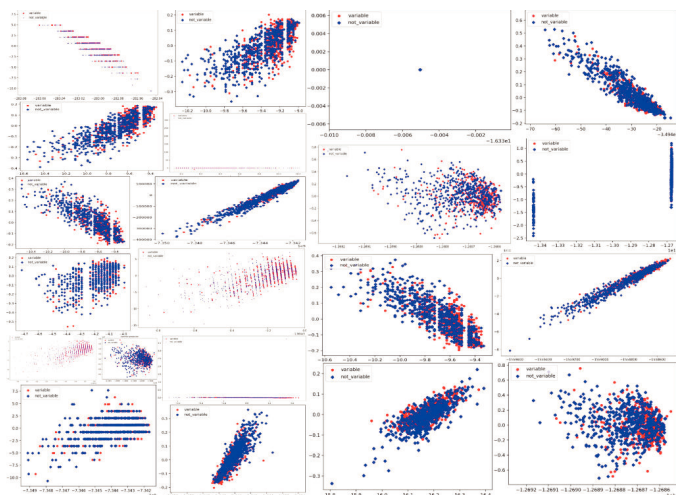


Fig. 6: Sammon mapping: 500 epochs, time series length: 1200-1225. Attr.: stand. dev., variance, min, max, mean, sum val., median, abs. sum of changes, agg. autocorr., arithmetic coeff., binned entropy, energy ratio, agg. FFT val., first loc. of min/max, mult. max values, index mass quant., linear trend etc. Or processed by FT or min-max normalization.

and non-variable objects. To verify this, Sammon projection was used (see Fig. 5 and 6). The experiment so far confirms that the original data does not contain clusters and we will have to focus on domain-specific details of the data in the future.

VI. CONCLUSION

Unfortunately, we did not get the accuracy over 65% using our current data pre-processing methods. Changing ANN types

or their parameters seems to have very little effect at results although we can see a positive affect of some act. functions. We believe that the major issue is improper data pre-processing that is unable to clearly distinguish variable and non-variable lightcurves for machine learning. One of the reasons might be the length of each light curve that is used as a feature vector. Therefore, we were (and still are) looking for better way of **dimensionality reduction using significant attributes extraction**. Based on ideas in [17], we believe that e.g. selecting n greatest or lowest **DFT coefficients** could be the way how to achieve this. By doing so, we should get a fixed-sized feature vector more suitable for machine learning.

In future, we plan to focus on the data in a more domain-specific way: try to distinguish concrete variability types from non-variable objects instead of all variables together. Then identify the best-performing activation functions and attempt to find better ones by means of **evolutionary algorithms**.

REFERENCES

- [1] B. Hu, Y. Chen, and E. Keogh, "Time Series Classification under More Realistic Assumptions," in Proceedings of the 2013 SIAM International Conference on Data Mining, Philadelphia, PA: Society for Industrial and Applied Mathematics, may 2013, pp 578–586.
- [2] Petitjean, F., Forestier, G., Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014). Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In 2014 IEEE International Conference on Data Mining. IEEE. <https://doi.org/10.1109/icdm.2014.27>
- [3] KARIM, F., MAJUMDAR, S., et al.: LSTM Fully Convolutional Networks for Time Series Classification, In: IEEE Access, vol. 6, 2017, pp. 1662 - 1669
- [4] CUI, Z., CHEN, W., CHEN, Y.: Multi-Scale Convolutional Neural Networks for Time Series Classification, In: arXiv:1603.06995v4, 2016
- [5] Jeffrey L. Elman. 1990. Finding Structure in Time. Cognitive Science 14, 2 (1990), 179–211. https://doi.org/10.1207/s15516709cog1402_1
- [6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (Nov. 1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [7] Hochreiter S. 1991. Untersuchungen zu dynamischen neuronalen Netzen. Master's thesis. TU Munich.
- [8] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. 2001. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. (2001).
- [9] C. Lee Giles, Steve Lawrence, and Ah Chung Tsoi. 2001. Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference. Machine Learning 44, 1 (01 Jul 2001), 161–183. <https://doi.org/10.1023/A:1010884214864>
- [10] LeCun, Y., BENGIO, Y.: Convolutional Networks for Images, Speech, and Time-Series
- [11] SAMUS, N., et al.: General Catalogue of Variable Stars: Version GCVS 5.1, In: Astronomy Reports, vol. 61, 2017, No. 1, pp. 80-88
- [12] ARMSTRONG, D., et al.: K2 Variable Catalogue I: A Catalogue of Variable Stars from K2, In. eprint arXiv:1411.6830, 2014
- [13] ARMSTRONG, D., et al.: K2 Variable Catalogue: Variable Stars and Eclipsing Binaries in K2 Campaigns 1 and 0. In: Astronomy and Astrophysics, 2015
- [14] THOMPSON, S., et al.: Kepler: A Search for Terrestrial Planets, 2016
- [15] JENKINS, M.: Kepler data preprocessing handbook, 2017
- [16] Sammon, J. W. (1969). A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers, C-18(5), 401–409. <https://doi.org/10.1109/t-c.1969.222678>
- [17] Wu, Y.-L., Agrawal, D., & El Abbadi, A. (2000). A comparison of DFT and DWT based similarity search in time-series databases. In Proceedings of the ninth international conference on Information and knowledge management - CIKM '00. ACM Press. <https://doi.org/10.1145/354756.354857>

Evaluation and analysis of bio-inspired optimisation algorithms for feature selection

Dražen Bajer, Bruno Zorić, Mario Dudjak, Goran Martinović

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

Josip Juraj Strossmayer University of Osijek, Osijek, Croatia

{drazen.bajer, bruno.zoric, mario.dudjak, goran.martinovic}@ferit.hr

Abstract—Wrapper-based feature (subset) selection is widely used as an effective means for decreasing the dimensionality of datasets. However, it is not the most efficient approach in terms of computational cost. Hence, the choice of the wrapper is paramount. Ideally, the wrapper should be simple to use and understand, whilst yielding good solutions as fast as possible. Bio-inspired optimisation algorithms are a common choice in that regard, but not all are made equally. This paper investigates a number of optimisers on diverse datasets in order to provide an insight into their efficiency and behaviour with respect to the problem of dimensionality reduction for classification needs. Correspondingly, some guidelines concerning the choice of the wrapper are given.

Index Terms—bio-inspired optimisation, classification, dimensionality reduction, feature selection, wrapper model

I. INTRODUCTION

With the advent of ever-larger datasets, both in terms of sample and feature count, the analysis and modelling of such data have become progressively more difficult. The large number of features is arguably one of the more prominent factors associated with this difficulty since not all are relevant and useful. In many instances the dimensionality reduction that can be achieved by removing redundant and irrelevant features is essential not only for creating simpler models with less computational resources, but also for increasing model accuracy and/or interpretability. Accordingly, feature subset selection or simply feature selection (FS) has become an almost unavoidable pre-processing step for many tasks in machine learning, such as classification, regression and clustering. The FS task can be performed in a variety of ways and a number of approaches have been proposed (see, e.g., [1], [2] for an overview). It is a valuable tool with applications spanning a wide range of problem domains where machine learning is employed for data analysis and as decision support. For example, a general overview can be seen in [3], whilst an overview in the highly important area of medicine can be seen in [4].

The aforementioned increase in dataset sizes simultaneously made the FS task grow more challenging. A complete enumeration of all $2^m - 1$ subsets is infeasible even for a moderate feature count (m) [1], [3]. Hence, there is a need for efficient search strategies. Several review or survey studies (see, e.g., [1], [3], [5]) identified evolutionary and swarm

intelligence algorithms (can be regarded as subsets of bio-inspired optimisation) as promising approaches for obtaining feature subsets since they can effectively explore large portions of the search space ($\mathcal{S} = \{0, 1\}^m$ for FS) and do not assume any specific properties of that space. A multitude of such algorithms have been proposed and many of them have been (successfully) applied to FS. Most of those algorithms are used to compose the wrapper model [5], which usually yields smaller and better subsets (in terms of model performance) than the filter model, but is typically computationally more expensive [2], [4].

The research on bio-inspired (or nature-inspired) optimisation is extremely active [6] and new algorithms are proposed on an almost daily basis. Many claim or promise superior performance and ease of use (few user-defined parameters, simple algorithm structure, etc.). This raises the question whether some approaches should be preferred. Accordingly, the performance and behaviour of several such algorithms are investigated on the problem of feature selection. More specifically, wrapper-based FS for classification needs was tackled with well-known, proven algorithms and some more recent ones. The experimental comparison was conducted on diverse datasets in order to offer an insight into the above-mentioned. In that regard, several indicators, like the attained solution quality, size and stability, have been considered. This should provide a glimpse into their differences but also into their advantages and shortcomings, at least with respect to FS.

The rest of the paper is organised as follows. Serving as introductory notes, a brief overview of feature selection and bio-inspired optimisation is given in Sect. II. In Sect. III two important elements are considered that need special attention when bio-inspired optimisers are applied to the problem at hand. The setup and the analysed results obtained in the experiments are reported and discussed in Sect. IV. Finally, the drawn conclusions and some avenues for future work are given in Sect. V.

II. BACKGROUND

A. Feature selection

Handling data described by a large number of features is difficult from various perspectives. Unfortunately, a large number of features does not imply a good description of the considered phenomenon since unnecessary (irrelevant and redundant) features are usually present [7]. Roughly speaking,

This research has been supported by the European Regional Development Fund under the grant KK.01.2.1.01.0127.

the FS task is to remove such features. In numerous instances, however, the features cannot be examined independently due to their interactions. These can vary in complexity, meaning that a useless single feature in combination with one or more others could become significant for the target concept [8].

The various methods available for approaching the FS task are traditionally categorised, based on the evaluation procedure, as wrappers and filters. Filters rely on intrinsic data properties while wrappers are guided by model performance. Although this makes the latter computationally more expensive and prone to overfitting, the cost is usually justified as they are able to discern complex relationships between features and generally lead to better and smaller subsets.

B. Bio-inspired optimisation

Generally, bio-inspired algorithms represent population-based global optimisation methods. They generally employ some form of variation operators on population members (candidate solutions) to create new solutions i.e. to sample the search space and, usually, selection to drive the search towards promising regions of that space. They can be viewed as stochastic, derivative-free methods and rely solely on the objective function values associated with points of the search space (black-box optimisation). As such, they are well suited for exploration, but less so for exploitation (at least compared to local search).

Amongst others, one of the best-known bio-inspired optimisers are genetic algorithms (GAs) [9], particle swarm optimisation (PSO) [10] and differential evolution (DE) [11]. However, the list of bio-inspired algorithms available in the literature goes a lot further and is immense (see, e.g., [12] for a comprehensive albeit incomplete list). The trend of proposing new optimisers is clearly apparent in the literature. Often peculiar metaphors are introduced to describe the search mechanism(s). Naturally, this approach to algorithm design and development (referred to as metaphor-centric in [13]) gained some criticism (see, e.g., [6], [13]–[15]) since the majority claims to be “novel” and “superior” in many aspects to previous ones, without offering proper supporting evidence. Nevertheless, a lot grew popular and have found diverse applications. Certainly, there are significant distinctions amongst many of those algorithms, which are mostly reflected in the operators or mechanisms for creating new solutions i.e. for sampling the search space. Interestingly, the vast majority of new proposals have operators defined in the continuous domain (\mathbb{R}^m). These operators typically do not use dedicated probability distributions for generating perturbations but use scaled differences between available candidate solutions kept as population members or as separate entities (popularised by PSO and DE). Further, from the viewpoint of a practitioner, the number of user-defined algorithm parameters is important since tuning is typically necessary for attaining best performance on the problem at hand. In that regard, fewer parameters might be considered a better option. There is a number of algorithms (more recent ones) that have a single user-defined parameter, the population size, but include some

intrinsic/internal parameters that are usually fixed, randomly generated or dynamically adjusted (see, e.g., [16]–[18]). However, this might be limiting in terms of flexibility/versatility compared to algorithms with multiple tunable parameters.

III. BIO-INSPIRED OPTIMISATION FOR WRAPPER-BASED FEATURE SELECTION

Tackling any problem with bio-inspired optimisation requires that an appropriate solution representation and evaluation criterion is chosen. Different choices for both can be found in the literature.

A natural solution representation is a binary vector $\mathbf{b} = (b_1, \dots, b_m) \in \{0, 1\}^m$, where $b_i = 1$ or 0 for $i = 1, \dots, m$ indicates that the i -th feature is or is not selected, respectively. Although this representation fits GAs, it does not fit the majority of other algorithms that have operators defined in the continuous domain. Hence, a transformation to obtain binary vectors is necessary. A simple and widely used approach is to use a sigmoid function (e.g. the logistic function) followed by thresholding. The purpose of the sigmoid function is to map the solution components to the same interval (typically $[0, 1]$ or $[-1, 1]$), but this may also be achieved by introducing a predefined boundary on the search space (e.g. $[0, 1]^m$) along with bound constraint handling (BCH). The BCH method is an additional design variable that can influence algorithm behaviour. It must be noted that these binary vectors are only generated prior to evaluation and do not replace the original, real-valued solution vectors. A few other representations can be found in the literature (see, e.g., [5]), but some require that the number of selected features is fixed a priori.

In the wrapper model, the classifier performance is used as the quality metric for found feature subsets. The classifier is used as a black-box and may be replaced by any other classifier of choice. However, there is a number of measures available for quantifying the classifier performance. The common choice is the classification accuracy (CAC) or conversely, the misclassification-rate (MCR). Often a convex combination of this measure and a penalty term (typically the normalised subset cardinality/size) is employed for the evaluation of subsets (see, e.g., [19]–[21]). This represents a simple approach to the treatment of FS as a bi-objective problem (these two objectives are not always conflicting [5]), where the contribution of the model performance and subset size is determined by the pre-set weight. Determining this weight is a major issue of that approach to multi-objective optimisation (MO) and is subjective. Naturally, MO bio-inspired algorithms that search for an approximation of the Pareto frontier can be found in the literature (see, e.g., [22], [23]) – they are well suited for such an optimisation since they maintain a population of solutions. It is important to remark that the CAC or MCR as a measure of classification performance can be misleading in the case of class imbalance [24], which could subsequently lead to the loss of features relevant to the minority class(es). Hence, a more sensible measure, like the F-score or the geometric mean of trues, might be better suited [25] and arguably preferred.

TABLE I
CONSIDERED BIO-INSPIRED OPTIMISATION ALGORITHMS

Algorithm	version	parameters	notes
ABC [26]	standard artificial bee colony (following the implementation in [27])	<ul style="list-style-type: none"> ○ population size $SN = 30$, ○ abandonment threshold $limit = 250$ 	Three separate search phases inspired by the foraging behaviour of three types of bees; Single solution components are updated at a time; New/updated solutions are accepted if they improve upon old/previous;
DE [11]	canonical differential evolution (DE/rand/1/bin)	<ul style="list-style-type: none"> ○ population size $NP = 50$, ○ scale factor $F = 0.5$, ○ crossover-rate $CR = 0.9$ 	Mutation is the main operator, with perturbations generated as scaled differences of selected population members (hence the name); Crossover is an auxiliary operator; Offspring compete for survival only with their respective parents;
GA [9]	simple/canonical genetic algorithm	<ul style="list-style-type: none"> ○ population size $N = 50$, ○ crossover probability $p_c = 0.9$, ○ mutation probability $p_m = 0.1$ 	Crossover is the main operator; Mutation is an auxiliary operator; Offspring completely replace the parent population;
PSO [10]	global-best particle swarm optimisation with inertia weight	<ul style="list-style-type: none"> ○ population size $NS = 30$, ○ acceleration coefficients $c_1 = c_2 = 1.496$, ○ inertia weight $\omega = 0.7298$ 	Each particle (solution) is guided towards both its personal best and global best position influenced by its velocity. New/updated solutions are accepted if they improve upon old/previous;
SCA [16]	standard sine cosine algorithm	<ul style="list-style-type: none"> ○ population size $n = 30$ 	Solutions are updated by moving them towards or away from the best found solution, where the sine and cosine functions are employed to achieve a cycling effect; New/updated solutions are accepted regardless of quality;
WOA [17]	standard whale optimisation algorithm	<ul style="list-style-type: none"> ○ population size $n = 30$ 	Three solution update mechanisms are employed inspired by the humpback whale hunting behaviour; Solutions are moved towards the best found or randomly selected candidate solutions; New/updated solutions are accepted regardless of quality;
Jaya [18]	standard Jaya algorithm	<ul style="list-style-type: none"> ○ population size $n = 20$ 	Solutions are updated by simultaneously moving them towards the best solution and away from the worst solution of each iteration; New/updated solutions are accepted if they improve upon old/previous;

TABLE II
CHARACTERISTICS OF THE DATASETS COMPRISING THE TEST BED

A	name	#features	#samples	#classes
1	QSAR biodegradation	41	1055	2
2	Connectionist Bench	60	208	2
3	Hill-Valley	100	1212	2
4	Ionosphere	34	351	2
5	Sonar	60	208	2
6	Dermatology	34	358	6
7	Image Segmentation	19	210	7
8	Libras Movement	90	360	15
9	Musk (Version 1)	166	476	2
10	Parkinsons	22	195	2
11	Statlog (Vehicle Silhouettes)	18	846	4
12	LSVT Voice Rehabilitation	310	126	2
13	Urban Land Cover	147	675	9
14	Wine	13	178	3

IV. EXPERIMENTAL ANALYSIS

In order to assess the performance and suitability of some bio-inspired optimisation algorithms for the wrapper-based feature selection task, an experimental analysis was conducted on a test bed comprised of diverse datasets. The selected algorithms are concisely shown in Table I, where the GA and DE can be categorised as evolutionary algorithms, whereas the others as swarm intelligence algorithms. It should be noted that only standard algorithm variants have been considered since the goal was to offer an insight into the differences of their intrinsic algorithm components. The employed datasets are given in Table II, where all have been taken from the UCI repository [28], except A_5 which has been taken from the KEEL repository [29].

A. Methodology and setup

The only algorithm in Table I that operates in the binary search space is the GA. For the others, the solutions were constrained into $[0, 1]^m \subset \mathbb{R}^m$ and binary vectors were created prior to evaluation via thresholding i.e. for each real-valued vector \mathbf{v} a separate binary vector \mathbf{b} was created as

$$b_i = \begin{cases} 1 & \text{if } v_i < \theta, \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, m, \quad (1)$$

where the threshold was set to $\theta = 0.5$. Solution components outside the search space were reset to the nearest boundary value. Feature subsets were evaluated by the F1-score, attained after applying the 1-NN classifier (the same as in [30]).

For each algorithm and dataset combination 25 independent runs were performed. The termination criterion was the execution of the pre-set maximal number of function evaluations, $NFE_{s_{\max}} = 10000$ in order to enable a fair comparison. Population initialisation was conducted uniformly at random inside the whole search space ($\{0, 1\}^m$ for the GA and $[0, 1]^m$ for the other algorithms). Algorithm parameters have been chosen according to recommendations in the literature and are reported in Table I. Further, stratified holdout evaluation was employed. Accordingly, the standard split ratio of 0.5 : 0.25 : 0.25 was used for training, validating and testing, respectively. A single split was generated and used in all algorithm runs since the primary goal was to evaluate the performance of optimisers in terms of performance and stability. Feature scaling, via normalisation into the $[0, 1]$ range, was performed as a pre-processing step on each dataset in order to mitigate the influence of varying value ranges.

TABLE III
RESULTS IN TERMS OF THE AVERAGE AND MAXIMAL ACHIEVED F1-SCORE

A	ABC		DE		SGA		PSO		SCA		WOA		Jaya	
	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>	<i>avg±std</i>	<i>max</i>
1	0.78±0.02	0.81	0.77±0.02	0.81	0.78±0.01	0.80	0.77±0.02	0.81	0.76±0.02	0.79	0.76±0.02	0.80	0.77±0.02	0.81
2	0.86±0.04	0.92	0.88±0.03	0.94	0.86±0.03	0.92	0.86±0.04	0.93	0.82±0.05	0.91	0.83±0.05	0.93	0.87±0.03	<i>0.96</i>
3	0.58±0.01	0.61	0.60±0.01	<i>0.62</i>	0.58±0.02	0.62	0.58±0.01	0.61	0.59±0.01	0.62	0.59±0.02	0.62	0.59±0.01	0.62
4	0.83±0.03	0.91	0.85±0.02	0.89	0.84±0.03	0.90	0.83±0.04	0.90	0.85±0.03	0.90	0.85±0.04	0.92	0.84±0.03	0.90
5	0.81±0.05	0.89	0.82±0.04	0.89	0.81±0.04	0.89	0.80±0.05	0.89	0.79±0.06	0.91	0.80±0.04	0.85	0.82±0.05	<i>0.91</i>
6	0.94±0.03	0.99	0.96±0.02	0.99	0.96±0.02	0.98	0.95±0.03	0.98	0.92±0.04	0.97	0.95±0.02	0.99	0.95±0.03	<i>1.00</i>
7	0.88±0.03	0.91	0.90±0.04	<i>0.93</i>	0.87±0.02	<i>0.93</i>	0.87±0.03	0.93	0.89±0.05	<i>0.93</i>	0.88±0.04	<i>0.93</i>	0.89±0.03	<i>0.93</i>
8	0.74±0.02	0.79	0.75±0.02	0.80	0.73±0.02	0.78	0.75±0.03	0.80	0.74±0.02	0.78	0.74±0.03	<i>0.81</i>	0.75±0.02	0.79
9	0.84±0.02	0.87	0.87±0.03	0.92	0.85±0.02	0.89	0.85±0.02	0.88	0.81±0.05	0.88	0.84±0.03	0.90	0.86±0.02	0.89
10	0.92±0.03	0.97	0.92±0.06	<i>1.00</i>	0.91±0.04	0.95	0.89±0.05	0.97	0.93±0.05	<i>1.00</i>	0.91±0.03	0.97	0.92±0.05	<i>1.00</i>
11	0.71±0.02	<i>0.74</i>	0.68±0.03	0.71	0.71±0.02	<i>0.74</i>	0.68±0.03	<i>0.74</i>	0.67±0.02	0.71	0.67±0.03	0.73	0.68±0.02	0.73
12	0.77±0.06	0.87	0.78±0.06	0.90	0.76±0.06	0.87	0.77±0.05	0.89	0.75±0.06	<i>0.90</i>	0.76±0.06	<i>0.90</i>	0.77±0.04	0.85
13	0.74±0.02	0.79	0.77±0.02	0.80	0.74±0.02	0.79	0.75±0.02	0.79	0.76±0.02	0.82	0.75±0.03	0.81	0.76±0.02	0.80
14	0.95±0.04	<i>1.00</i>	0.95±0.03	<i>1.00</i>	0.95±0.03	<i>1.00</i>	0.95±0.03	<i>1.00</i>	0.95±0.03	<i>1.00</i>	0.94±0.03	0.98	0.96±0.03	<i>1.00</i>
FR	3.96		1.71		4.61		5.14		4.71		5.07		2.79	
d_2 (d_∞)	0.79 (0.42)		0.76 (0.40)		0.80 (0.42)		0.81 (0.42)		0.83 (0.41)		0.82 (0.41)		0.78 (0.41)	

TABLE IV
RESULTS OF THE WILCOXON TEST FOR MULTIPLE COMPARISONS

i^a	ABC	DE	SGA	PSO	SCA	WOA	Jaya
ABC	—	⊙					⊙
DE	⊕	—	⊕	⊕	⊕	⊕	⊕
SGA		⊙	—				⊙
PSO		⊙		—			⊙
SCA		⊙			—		⊙
WOA		⊙				—	⊙
Jaya		⊙	⊕	⊕	⊕	⊕	—

^a ⊕ / ⊙ denotes that the algorithm in the row / column improves the one in the column / row; upper / lower diagonal represents a level of significance of $\alpha = 0.9 / \alpha = 0.95$.

B. Results and discussion

The obtained results in terms of classification performance on the test subsets are presented in Table III. In order to ease the determination of the mutual order of the competing algorithms in terms of performance, two distinct measures are provided at the bottom of the table (the same approach as used in [31]). The first one, denoted by FR represents the average rankings obtained by applying the Friedman test for multiple comparisons. The second one, denoted by d_2 , represents the Euclidean distance from the perfect classifier where also in parentheses the Chebyshev distance (d_∞) is provided for breaking ties. In both cases, a lower value is better. To facilitate readability, the best results on each dataset are highlighted in boldface and italic.

At first glance, DE stands out in terms of performance, followed closely by Jaya. This is put into perspective by the results shown in Table IV, where the superiority of DE and Jaya is obvious since they were the only ones that performed better than the remaining competitors in a statistically significant manner. It is worth noting that DE improves even upon Jaya in a statistically significant manner. The performance of DE is not that surprising, considering its effectiveness on a myriad of problems demonstrated in the literature. What may come as a surprise is the good overall performance of Jaya. Despite

being a relatively recent, comparatively unproven algorithm, it was still able to come close to the performance of a well-established one. This suggests its search mechanism to be effective, at least in the case of the problem at hand. The differences amongst the remaining optimisers are not as clear-cut and warrant a more in-depth look at the performance metrics. When considering the average rankings (FR), two groupings are visible. Inside the worse performing group, consisting of ABC, SGA, PSO, SCA and WOA, no compelling differences are apparent. However, an insight into the potential differences amongst these is offered by the provided distance measures, enabling a sub-ordering, where WOA and SCA came out behind the rest. Further, the standard ABC algorithm came out as the strong third contender. The overall lower performance compared to DE might be attributed to its inherently low convergence-rate due to the search mechanism that updates only a single solution component at a time. Presumably, this may become increasingly notable with a growth in problem dimensionality.

Another important aspect of the FS task are certainly the sizes of the selected feature subsets, results of which are reported in Table V. Presented are the ratios of the number of selected and total features in order to facilitate the interpretation of the results. These are also accompanied by the measure of stability which is indicative of the stability in finding feature subsets that are consistent across multiple runs. It is called the adjusted stability measure (ASM) [32] and was selected due to its suitability for comparing feature subsets of varying sizes. It must be noted that the above-mentioned measure has been scaled by 10^2 in order to paint the differences clearer. None of the algorithms included in the comparison is overwhelmingly stable, keeping in mind that the range of ASM lies in $[-1, 1]$. This is due to their stochastic nature and the fact that FS is a multimodal problem where different feature subsets can result in a virtually equal classification performance. A glimpse into this can be gained from Fig. 2. Nevertheless, they cannot be deemed unstable for that matter. A different perspective on

TABLE V
RESULTS IN TERMS OF AVERAGE SUBSET SIZE AND STABILITY

A	ABC		DE		SGA		PSO		SCA		WOA		Jaya	
	avg±std	ASM	avg±std	ASM	avg±std	ASM	avg±std	ASM	avg±std	ASM	avg±std	ASM	avg±std	ASM
1	0.47±0.08	0.12	0.47±0.09	1.08	0.48±0.08	1.12	0.49±0.07	0.44	0.24±0.05	1.76	0.50±0.14	0.84	0.48±0.08	0.36
2	0.48±0.06	0.66	0.47±0.05	1.28	0.47±0.06	0.32	0.47±0.06	0.26	0.20±0.06	0.19	0.35±0.11	0.02	0.46±0.06	0.60
3	0.46±0.05	0.39	0.45±0.04	1.46	0.47±0.05	0.51	0.47±0.05	0.43	0.09±0.02	0.41	0.13±0.06	0.21	0.45±0.05	0.72
4	0.40±0.08	1.05	0.33±0.08	1.74	0.36±0.08	0.68	0.41±0.09	0.91	0.19±0.04	1.33	0.27±0.13	0.44	0.41±0.08	0.86
5	0.49±0.06	0.28	0.47±0.07	0.72	0.46±0.06	0.26	0.50±0.07	0.10	0.20±0.04	0.41	0.36±0.17	0.46	0.47±0.06	0.44
6	0.57±0.09	0.38	0.60±0.08	1.07	0.63±0.05	1.31	0.59±0.08	0.29	0.35±0.08	0.83	0.69±0.14	1.40	0.61±0.06	1.02
7	0.38±0.09	3.68	0.26±0.07	4.34	0.33±0.09	4.11	0.40±0.13	1.90	0.24±0.09	3.69	0.36±0.15	1.66	0.42±0.13	2.68
8	0.47±0.06	0.14	0.45±0.06	1.11	0.48±0.04	0.41	0.47±0.05	0.53	0.18±0.04	0.44	0.36±0.11	0.19	0.46±0.05	0.67
9	0.49±0.04	0.10	0.46±0.03	0.51	0.49±0.04	0.18	0.47±0.04	0.19	0.12±0.05	0.69	0.35±0.16	0.24	0.48±0.04	0.36
10	0.49±0.10	0.76	0.43±0.11	1.61	0.45±0.09	1.58	0.53±0.08	0.72	0.36±0.11	1.38	0.50±0.14	0.69	0.47±0.12	0.71
11	0.47±0.08	2.85	0.46±0.10	1.88	0.49±0.07	2.33	0.48±0.10	1.54	0.35±0.06	2.54	0.43±0.11	1.42	0.53±0.11	1.54
12	0.50±0.03	0.06	0.50±0.03	0.27	0.49±0.03	0.13	0.49±0.03	0.05	0.22±0.08	0.17	0.36±0.19	0.12	0.50±0.03	0.07
13	0.49±0.05	0.18	0.48±0.04	0.67	0.50±0.04	0.22	0.50±0.04	0.21	0.14±0.07	0.46	0.55±0.17	0.10	0.49±0.04	0.58
14	0.52±0.10	1.45	0.51±0.08	0.75	0.54±0.10	0.59	0.51±0.10	1.53	0.50±0.10	0.93	0.57±0.11	0.62	0.54±0.12	0.36

TABLE VI
AVERAGE RANKINGS FROM THE FRIEDMAN TEST DURING SEARCH

Alg.	Fraction of maximally allowed function evaluations used													
	0.01	0.02	0.03	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
ABC	6.71	6.57	6.54	6.50	6.43	6.36	6.29	6.14	6.00	5.82	5.54	5.11	5.00	5.04
DE	4.11	4.25	3.82	3.46	2.79	2.11	1.96	1.93	1.86	1.86	1.75	1.86	1.64	1.64
SGA	4.36	4.89	4.96	5.14	5.04	5.11	5.04	5.07	5.11	5.04	5.11	5.25	4.89	4.79
PSO	3.61	2.43	2.43	2.57	3.07	3.57	3.64	3.82	3.86	3.86	3.89	3.96	4.11	4.21
SCA	4.39	3.64	3.50	3.07	3.25	3.21	2.96	2.96	3.00	3.07	3.14	3.29	3.36	3.29
WOA	3.25	4.50	4.82	5.07	5.18	5.04	5.32	5.18	5.11	5.36	5.39	5.54	5.57	5.71
Jaya	1.57	1.71	1.93	2.18	2.25	2.61	2.79	2.89	2.96	3.04	3.07	3.11	3.21	3.32

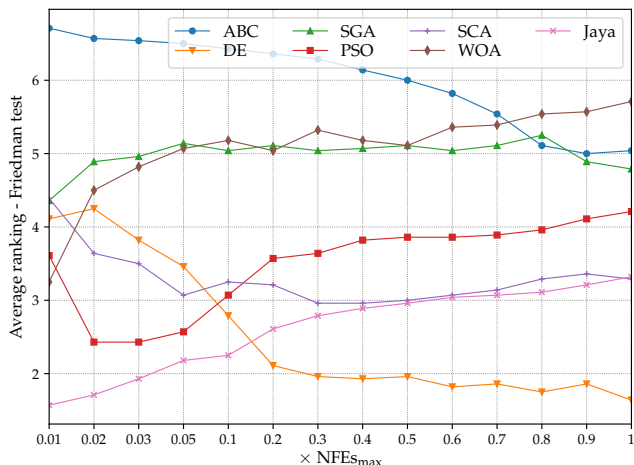


Fig. 1. Illustration of mutual algorithm ordering throughout the search

algorithm stability is offered by the standard deviation of the results in terms of the classification performance (average F1-score) and the average sizes of the attained subsets. These results suggest that the competing algorithms are relatively stable in terms of former but less in terms of the latter. The obtained feature subsets of varying sizes are in accordance with the problem multimodality. The inclusion of the subset size into the objective function is a way of combating this issue, as stated earlier, but the primary focus of the search

should still be classification performance. When considering the found subset sizes, it can be seen that the largest reductions were achieved by the two worst performing algorithms in terms of classification performance (SCA and WOA). All other algorithms found subsets of similar average sizes.

It is interesting to observe that while binary solution encoding is suitable for the FS problem, the only algorithm operating directly with binary solutions, namely SGA, was outperformed by several other algorithms that operate on real-valued solutions and later transform them into binary space. A brief post-hoc experiment indicated that the incorporation of elitism in the form of the $\mu + \lambda$ survival selection procedure (with $\mu = \lambda = N$) resulted in a significant performance boost, reaching almost that of DE [as suggested by $(d_2, d_\infty) = (0.77, 0.40)$]. This is certainly promising since further improvements in performance may be expected with the replacement of the variation operators and an in-depth investigation is warranted in order to reveal its potential. Also interesting is that over half of the considered algorithms utilise solution updating mechanisms that incorporate a movement towards the best-so-far solution. However, as is apparent from the presented results, not all are made equally. The one utilised in Jaya seems to be more effective since it performed notably better than the others. In the end, the performance of PSO might also come as surprise, especially since it has been used as the foundation for numerous FS approaches (see, e.g., [5]). As in the case of the SGA, a brief post-hoc experiment was conducted. The obtained results suggested that the chosen

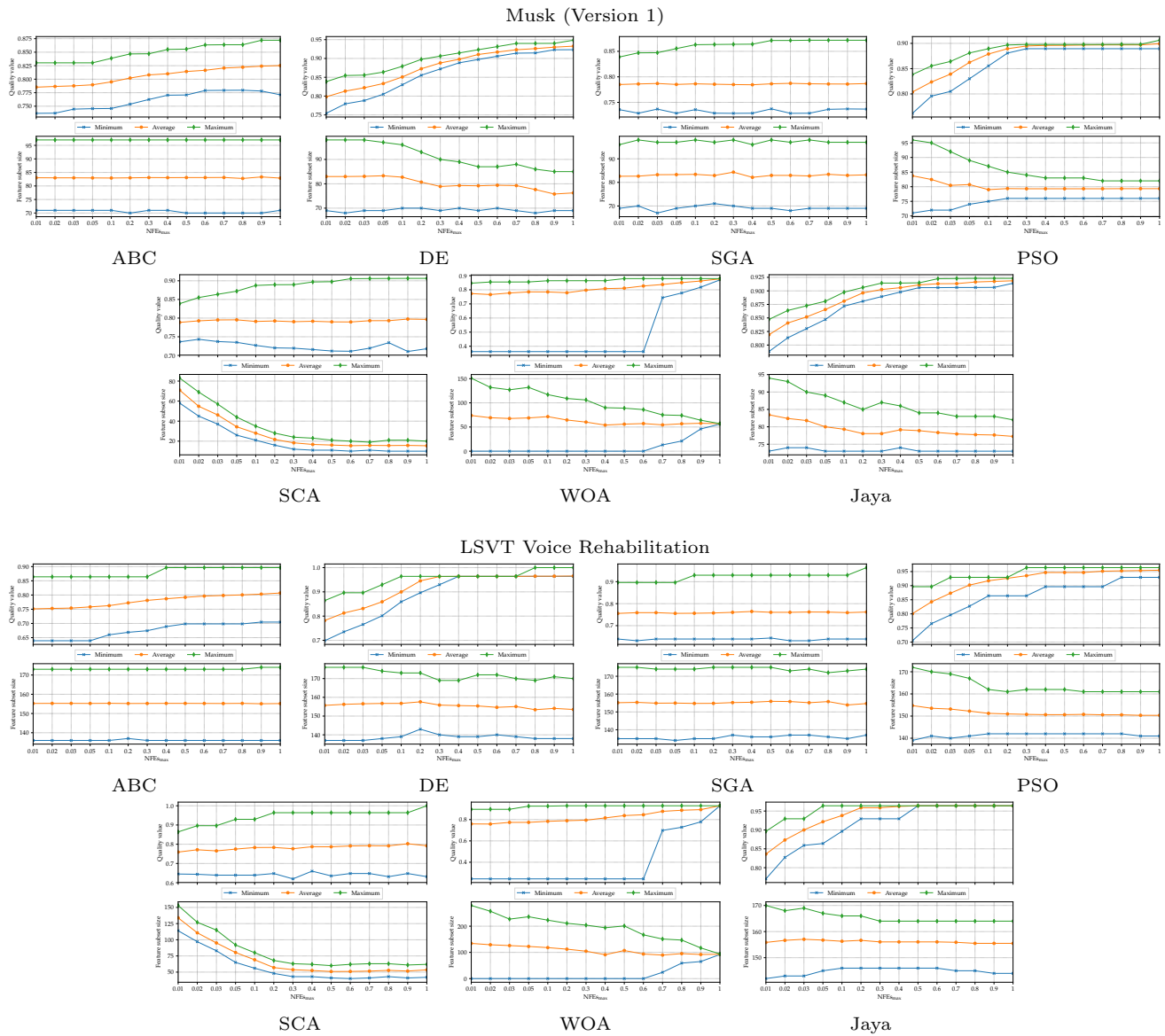


Fig. 2. Convergence graphs (median runs) for each algorithm on two selected datasets (A_9 and A_{12}). The upper subfigure shows the minimum, average and maximum values of the F1-score throughout the search process whilst the lower subfigure shows the same for the size of found subsets.

parameter settings might be responsible, although they are widely used in the literature. Changing the parameters to $\omega = 0.729$, $c_1 = 2.0412$ and $c_2 = 0.9477$ resulted in a notable performance improvement as indicated by $(d_2, d_\infty) = (0.79, 0.42)$. It can be expected that some further boosts might be obtained through proper parameter tuning.

Algorithm behaviour during the search process is shown by the convergence graphs for two datasets in Fig. 2. The behaviour on only two datasets is presented in the figure to showcase the differences during search due to space constraints, although similar behaviour was observed on others as well. Additionally, average rankings during search, obtained by applying the Friedman test across all datasets, are given in Table VI. To facilitate interpretability, the same results are presented in Fig. 1. Notably, Jaya has an overall greater

convergence-rate the others in the early search phases. It can be assumed that this is due to its solution update mechanism where it moves the solution towards the current best and away from the current worst. However, as is visible from the overall average rankings (Friedman test), at the later stages of the search it is unable to sustain this momentum, while DE is still able to improve thus achieving the best ranking at the end. The fact that DE achieved the best rankings overall both on the validation and testing data is suggestive of the lack of overfitting. Accordingly, an excessive exploitation might be a reasonable approach only in the case of a restricted number of function evaluations and/or low dimensional datasets. A similar behaviour can also be achieved in DE by incorporating a mutation operator that is focused on exploitation (like best/1). Nevertheless, a proper exploration ability seems to be

of considerable importance for obtaining good solutions. This is also supported by the post-hoc experiment with PSO, where increasing the emphasis on exploration (by setting $c_1 > c_2$) resulted in a performance improvement. Achieving a similar effect might be, however, difficult with the more recent of the considered algorithms since their search mechanisms are "fixed" and only a single user-defined parameter is exposed (the population size), whilst the introduction of additional mechanisms would produce hybrids. Nevertheless, in order to gain a glimpse into this matter, a post-hoc experiment was conducted with SCA and WOA, where the population size was increased to $n = 50$ with the aim of increasing their exploration ability. In the case of SCA only a slight improvement was observed as indicated by $(d_2, d_\infty) = (0.82, 0.41)$, whereas in the case of WOA the performance improvement was more substantial, according to $(d_2, d_\infty) = (0.80, 0.41)$. Yet, these improvements have not led to a change of the mutual ordering amongst the considered algorithms in terms of overall performance. Further, Table VI also provides support for the earlier statement about the slow convergence of ABC. In the aforementioned figures, WOA stands out. Its peculiar behaviour illustrates the solution update mechanism behind the algorithm, where gradually solutions closer to the best-so-far are generated until the whole population converges in the end. It was also the only one among the considered algorithms that exhibited convergence of the population, although it does not incorporate a selection procedure and new solutions are accepted regardless of quality. This plays probably a notable role in the competitively low performance it attained since exploration is impaired in the later phases of the search.

V. CONCLUSION

In this paper the behaviour of several bio-inspired optimisation approaches was explored for the feature selection problem. Several well-known approaches along with a few more recent ones were selected and experimentally evaluated on diverse datasets. Based on the presented results obtained in the experimental analysis, the DE algorithm can be recommended for tackling the FS task. The feature subsets it attained were not the smallest but resulted in the overall highest performance both in terms of classification performance and stability. Another favourable aspect of DE is that numerous enhancements (e.g. a myriad of mutation operators is readily available) and improved variants can be found in the literature. Another reasonable suggestion is the GA. The simple GA (SGA) did not perform nearly as good as DE, however, the preliminary examination revealed promising results attained by a simple replacement of the survivor selection procedure and further improvements may be expected with the replacement of the variation operators (similar to the case of DE, numerous operators are readily available). The main selling point of the GA is that it directly operates in the binary solution space and thus no transformation of candidate solutions is necessary. Apart from Jaya, which was able to compete with DE, the other included recent optimisers (SCA and WOA) were not able to do so and cannot be recommended.

Although run-to-run consistency of found subsets (as measured by stability) is rarely considered when bio-inspired optimisers are applied to FS, it is something that should not be taken lightly. Namely, it is of paramount importance for the interpretability of feature subsets as well as for providing proper insight into feature interactions and relevance. Accordingly, appropriate measures should be taken as to ensure relative stability when developing algorithms for FS. A number of approaches that can be considered are presented in [33]. This is where DE again performed overall better than the other competing optimisers, albeit neither can be regarded particularly stable.

One possible direction for future work could be a more extensive analysis including other bio-inspired optimisers as well as larger datasets, both in terms of sample and feature count. The latter represents a more serious issue and should be kept in mind since, according to the literature (see, e.g., [5]), these approaches are not suitable when dealing with several thousand features. Pre-processing steps (like an initial reduction via filter-based approaches or clustering) could aid in this case and warrant further consideration. Further, the employed parameter values were taken as recommended by the literature and represent the approach likely to be taken by an aspiring practitioner. However, the question could be raised about the gains in performance that could be obtained through parameter tuning. Moreover, the number of tunable algorithm parameters is supposed to play an important role in this regard. For instance, to which extent is the number of parameters a limiting factor? In that regard, DE again seems like a better choice than Jaya, which uses randomly generated scales inside a fixed range. This may still be considered less limiting than the parameters "hidden" inside SCA and WOA.

Finally, this paper might be regarded as another critique on the metaphor-centric algorithm development approach. Hence, it is worth noting that this was not the intention. However, the presented results do not provide any compelling reasons for favouring them over the much older algorithms. On the contrary, the well-established algorithms proved to be the reasonable choice for developing FS approaches.

REFERENCES

- [1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.
- [2] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomput.*, vol. 300, pp. 70–79, 2018.
- [3] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. MIPRO'15*, 2015, pp. 1200–1205.
- [4] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," *Comput. Biol. Med.*, vol. 112, p. 103375, 2019.
- [5] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, 2016.
- [6] J. D. Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, and F. Herrera, "Bio-inspired computation: Where we stand and what's next," *Swarm Evol. Comput.*, vol. 48, pp. 220–250, 2019.
- [7] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.
- [8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

- [9] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2015.
- [10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. ICEC'98*, 1998, pp. 69–73.
- [11] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [12] J. Rajpurohit, T. K. Sharma, A. Abraham, and Vaishali, "Glossary of metaheuristic algorithms," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 9, pp. 181–205, 2017.
- [13] K. Sörensen, M. Sevaux, and F. Glover, "A history of metaheuristics," in *Handbook of Heuristics*, R. Martí, P. Pardalos, and M. Resende, Eds., 2018.
- [14] M. A. Lones, "Metaheuristics in nature-inspired algorithms," in *Proc. GECCO'14*, 2014, pp. 1419–1422.
- [15] K. Sörensen, "Metaheuristics—the metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, 2015.
- [16] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, 2016.
- [17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.
- [18] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 7, no. 1, pp. 19–34, 2016.
- [19] G. Martinović, D. Bajer, and B. Zorić, "A differential evolution approach to dimensionality reduction for classification needs," *Int. J. Appl. Math. Comput. Sci.*, vol. 24, no. 1, pp. 111–122, 2014.
- [20] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomput.*, vol. 213, pp. 54–65, 2016.
- [21] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, 2019.
- [22] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Inf. Sci.*, vol. 422, pp. 462–479, 2018.
- [23] N. Kozodoi, S. Lessmann, K. Papakonstantinou, Y. Gatsoulis, and B. Baesens, "A multi-objective approach for profit-driven feature selection in credit scoring," *Decis. Support Syst.*, vol. 120, pp. 106–117, 2019.
- [24] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, 2013.
- [25] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [26] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.
- [27] M. Mernik, S.-H. Liu, D. Karaboga, and M. Črepinšek, "On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation," *Inf. Sci.*, vol. 291, pp. 115–127, 2015.
- [28] K. Bache and M. Lichman, "UCI machine learning repository," 2013, <http://archive.ics.uci.edu/ml>.
- [29] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *J. Mult.-valued Log. Soft Comput.*, vol. 17, no. 2-3, pp. 255–287, 2011.
- [30] B. Zorić, D. Bajer, and G. Martinović, "Utilising filter inferred information in nature-inspired hybrid feature selection," in *Proc. SST'18*, 2018, pp. 117–123.
- [31] D. Bajer, B. Zorić, M. Dudjak, and G. Martinović, "Performance analysis of SMOTE-based oversampling techniques when dealing with data imbalance," in *Proc. IWSSIP'19*, 2019, pp. 265–271.
- [32] J. L. Lustgarten, V. Gopalakrishnan, and S. Visweswaran, "Measuring stability of feature selection in biomedical datasets," in *Proc. AMIA'09*, vol. 2009, 2009, p. 406.
- [33] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ. - Comp. Inf. Sci.*, 2019.

Towards Automated Translating Natural Language Sentences into Intensional Constructions

Branislav Bednár

Student in Department of Computers and Informatics
Technical University of Košice
Košice, Slovakia

Zuzana Bilanová

Department of Computers and Informatics
Technical University of Košice
Košice, Slovakia

Abstract—This paper aims to introduce a new and original implementation of the semantic machine of transparent intensional logic. This logic system is characterized by its procedural semantics, thanks to which it is used in the field of logical analysis of natural language. Transparent intensional logic is a high-order logic, where partial functions of the intensional typed λ -calculus allow to remove the semantic ambiguities of natural language. Natural language expressions are usually analyzed in three steps - type analysis, synthesis of construction, and type control, which is an optional step. The presented implementation of the semantic machine of transparent intensional logic is focused on the first two of the mentioned steps of formalizing the meaning of sentences. This semantic machine uses outputs from the Stanford CoreNLP parser. Its functionality and correctness are demonstrated at the end of the paper through the analysis of several sentences of natural language.

Index Terms—logical analysis of natural language, natural language processing, semantic machine

I. INTRODUCTION

Foundations of transparent intensional logic (TIL) were created in 1961 by Czech logician Pavel Tichý [1]. TIL language is a revised version of the typed object-oriented λ -calculus with a ramified hierarchy of types. TIL object base contains a set of truth values o , a set of individuals ι , a set of time points τ and a set of possible worlds ω [2]. The intensions are objects of type $((\alpha\tau)\omega)$ - functions from possible worlds in time points for any type α .

Tichý suggested his own modification of well-known Frege's semantic triangle. Figure 1 shows that in Frege's approach expression expresses a sense and this determines a referent which denotes the expression. Tichý introduced an approach where a sense of expression is expressed by its construction.

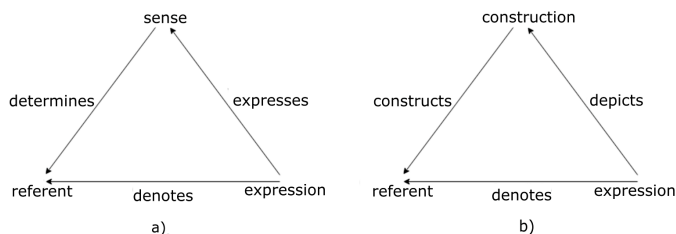


Fig. 1. Tichý's modification of Frege semantic triangle [3]. Figure a) shows Frege's interpretation of a semantic triangle and figure b) shows Tichý's modification.

Constructions are abstract procedures used to represent the meaning of natural language sentences. There are four basic kinds of constructions, similar to λ -terms: variable x , trivialization $\hat{0}A$ (corresponds to a constant), composition $[AA]$ (corresponds to an application), and closure λxA (corresponds to an abstraction) [4]. This logic can be also extended by linear logic [5], [6].

The competitive modal logic system [7] - Montague intensional logic (MIL) [8], have several drawbacks that TIL had overcome. Tichý's logical apparatus is anti-contextualist (all expressions are intensions), universal (MIL is bounded to the English language) and has a direct method of interpreting the meaning of an expression (MIL needs to translate the expression of the natural language into an intensional logical language which has to be interpreted). The fact that TIL is an effective tool for logical analysis of natural language, making it an ideal basis for creating a semantic machine.

II. SYNTACTIC ANALYZERS

Successful implementation of a semantic machine requires performing an analysis on two basic layers of natural language - syntactic and semantic [9]. The syntactic analysis allows creating an output used for the semantic analysis [10]. Since the presented research deals only with semantic analysis of natural language sentences, it was decided to use an existing tool for syntactic analysis. In this chapter, two of the most commonly used syntactic analyzers will be compared - Link Grammar Parser and Stanford CoreNLP.

A. Link Grammar Parser

Link Grammar Parser [11] is a syntax analyzer of the English language. The parser was written in C programming language and it is completely open-source. The size of the verbal apparatus is more than 60 000 words in different forms. In the Figure 2 an output from Link Grammar Parser for sentence „Peter is a student.” is shown. The output offers the display and naming of the links between individual words within a sentence. The tree of sentence components is also displayed by indenting the text. In the tree of sentence components, each word is assigned a word class. The disadvantage is that the output is only in plain text form, so it is hard to parse this output and get some results from it.

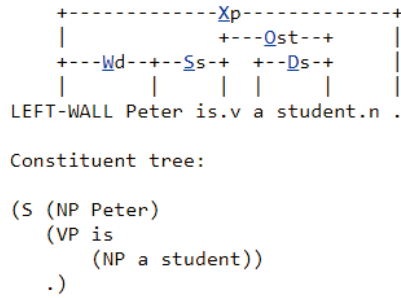


Fig. 2. Analysis of the sentence „Peter is a student.” generated from Link Grammar Parser. The meaning of the used symbols: S - sentence, NP - noun phrase, VP - verb phrase, Xp - periods at ends of sentences, Wd - attaches main clauses to the wall, Ost - connects transitive verbs to direct or indirect objects, Ss - connects subject-nouns to finite verbs, Ds - connects determiners to nouns, read more at [11].

B. Stanford CoreNLP

Stanford CoreNLP [12] comes as a more complex and sophisticated open-source parser. It is an annotation-based natural language processing framework. In addition to English, this parser also supports other languages, such as Arabic, Chinese, French, German, Spanish. The parser is written in Java programming language and the big advantage is the interfaces it offers. Thanks to them, it is possible to use CoreNLP with languages as C#, JavaScript, Python or Scala and it is also compatible with Docker. The parser can read a single sentence, as well as a block of text in its input. If more sentences are used as an input, then every sentence is analyzed and Stanford CoreNLP returns a collection of analyzed sentences in the output.

Stanford CoreNLP offers more opportunities to further data processing than Link Grammar analyzer (illustrated in Figure 3). It returns the result of sentence analysis in json, which is easy to parse.

The most useful part of its output for our implementation of semantic machine is *basicDependencies* where are defined dependencies between words and *tokens* where it is possible to find a part of speech (POS Tagger) for every word in our sentence. Based on advantages shown in Table I it was decided to use Stanford CoreNLP as the basis of the TIL semantic machine.

TABLE I
LINK GRAMMAR PARSER AND STANFORD CORENLP COMPARISON

	Link Gram- mar parser	Stanford CoreNLP
API	-	+
multilanguage	-	+
open source	+	+
editable	+	+
output visualization	-	+
last released version	2005	2018
parser configuration	-	+
text analysis	-	+
output format configuration	-	+

```

{tokens:
[{'after': '',
'before': '',
'characterOffsetBegin': 0,
'characterOffsetEnd': 5,
'index': 1,
'lemma': 'Peter',
'originalText': 'Peter',
'pos': 'NNP',
'word': 'Peter'},
{'after': '',
'before': '',
'characterOffsetBegin': 6,
'characterOffsetEnd': 8,
'index': 2,
'lemma': 'be',
'originalText': 'is',
'pos': 'VBZ',
'word': 'is'},
{'after': '',
'before': '',
'characterOffsetBegin': 9,
'characterOffsetEnd': 10,
'index': 3,
'lemma': 'a',
'originalText': 'a',
'pos': 'DT',
'word': 'a'},
{'after': '',
'before': '',
'characterOffsetBegin': 11,
'characterOffsetEnd': 18,
'index': 4,
'lemma': 'student',
'originalText': 'student',
'pos': 'NN',
'word': 'student'}]]

[{'basicDependencies':
[{'dep': 'ROOT',
'dependent': 4,
'dependentGloss': 'student',
'governor': 0,
'governorGloss': 'ROOT'},
{'dep': 'nsubj',
'dependent': 1,
'dependentGloss': 'Peter',
'governor': 4,
'governorGloss': 'student'},
{'dep': 'cop',
'dependent': 2,
'dependentGloss': 'is',
'governor': 4,
'governorGloss': 'student'},
{'dep': 'det',
'dependent': 3,
'dependentGloss': 'a',
'governor': 4,
'governorGloss': 'student'}]]

[{'enhancedDependencies':
[{'dep': 'ROOT',
'dependent': 4,
'dependentGloss': 'student',
'governor': 0,
'governorGloss': 'ROOT'},
{'dep': 'rsubj',
'dependent': 1,
'dependentGloss': 'Peter',
'governor': 4,
'governorGloss': 'student'},
{'dep': 'cop',
'dependent': 2,
'dependentGloss': 'is',
'governor': 4,
'governorGloss': 'student'},
{'dep': 'det',
'dependent': 3,
'dependentGloss': 'a',
'governor': 4,
'governorGloss': 'student'}]]]

```

Fig. 3. Analysis of the sentence „Peter is a student” generated from Stanford CoreNLP, read more about *tokens*, *basicDependencies* and *enhancedDependencies* at [12].

C. Definition of dependencies

Since it was decided to use a third-party syntax analyzer, it is necessary to understand data in its output. For the implementation of the semantic machine, it is important to focus on *basicDependencies* and the last part of the output, where all the words are described. There are some selected dependencies explained [13]:

- nsubj - nominal subject,
- cop - copula,
- neg - negation modifier,
- dobj - direct object,
- aux - auxiliary,
- advmod - adverb modifier,
- amod - adjectival modifier,
- det - determiner,
- conj - conjunct,
- poss - possession modifier,
- nsubjpass - passive nominal subject,
- nmod - nominal modifier.

III. SEMANTIC MACHINE OF TRANSPARENT INTENSIONAL LOGIC

After syntactic analysis, semantic analysis of sentences presents the second step of the natural language processing. In case of using TIL principles, the output of the semantic analysis is a sentence in the form of TIL constructions. Those constructions are suitable for next processing.

Until now, it was created only one implementation of the TIL - its author is Aleš Horák [3]. His semantic machine presents a solution for the Czech language and covers almost the entire conceptual apparatus. In Horák's approach and also in our semantic machine, natural language sentence analysis is performed in three steps [14] (shown in Figure 4):

- 1) type analysis - to each object of the sentence is assigned its type,
- 2) creation of construction - construction represents the meaning of a sentence,
- 3) type control - verifying if the previous step was correctly done (not a necessary step).

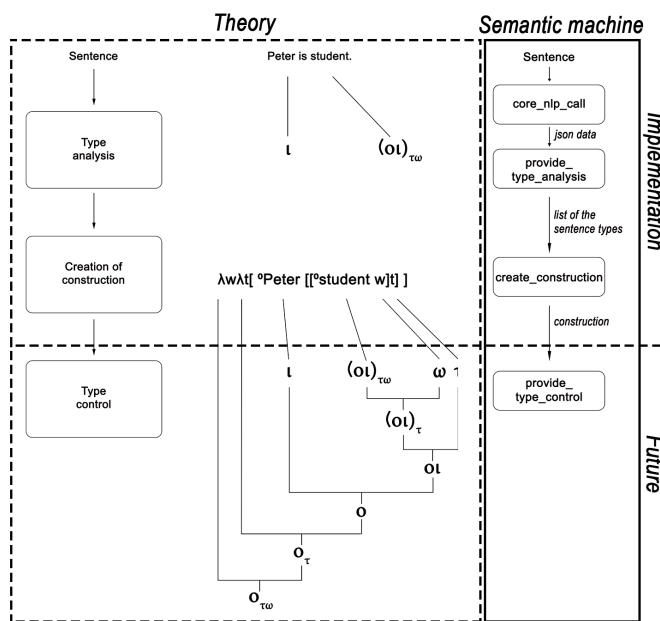


Fig. 4. The principle of three-step sentence analysis in TIL. The left side of the Figure presents an analysis of the sentence: "Peter is a student." in TIL, while the right side of the Figure describes how the sentence is processed by our semantic machine.

The right side of the Figure 4 shown input processing. The input is firstly sent to Stanford CoreNLP which provides syntactic analysis and return analyzed sentence in form of json. Then the semantic machine in the module *provide_type_analysis* makes a type analysis and saves all the recognized types into a list. The list should be passed to the module *create_construction*. After construction is successfully created, it is printed on the output of the semantic machine and the necessary data are supplied to the module *provide_type_analysis* (this is a subject of future research).

A. Type analysis

The first step of our implementation of the semantic machine is the type analysis. The analyzed types extend basic atomic TIL types o , ι , ω , τ .

Individual ι - presents one of the basic types which Tichý defined. The individual represents a specific entity that always

starts with a capital. Stanford CoreNLP indicates it as *NNP* - a proper noun, singular (shown in Figure 5).

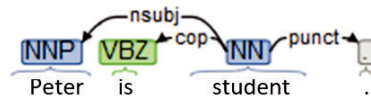


Fig. 5. Assign of individual type.

Truth value o - also presents one of the basic types in TIL. In the sentence „Sun is shining.” word *shining* represents truth value because it defines the state of the sun in some world and time. The truth value can be obtained as an independent component of the nominal subject *nsubj* (shown in Figure 6).

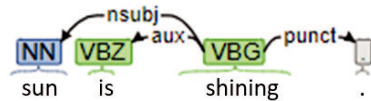


Fig. 6. Assign of truth value type.

Attribute $((\iota)\tau)\omega$ - presents an extended type. There is a need to find a dependency *case*, where an individual is the dependent component and the appropriation indicator 's represents the independent component. Then the dependency *nmod:poss* can be found - the dependent component is the individual and the independent component is the attribute which is owned by the individual. If there is no individual in the sentence, it is necessary to find *nmod:poss* dependency directly (shown in Figure 7).

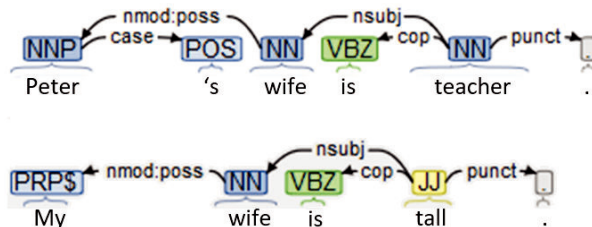


Fig. 7. Assign of attribute type.

Property of the object $((oi)\tau)\omega$ - presents an extended type. To determine this type, there should be a dependency *nsubj*, where an individual is the dependent component and the truth value is the independent component. E.g., in the sentence „Bob is student.” the property *student* is assigned to the individual *Bob* (shown in Figure 8).

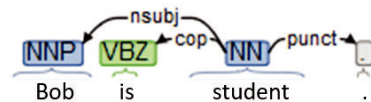


Fig. 8. Assign of property of object type.

Property of the property $((((oi)\tau)\omega)((oi)\tau)\omega)$ - presents an extended type. To determine this type, there should be a

dependency *amod*, where the dependent component presents property of property and the independent component is property assigned to the object (shown in Figure 9).

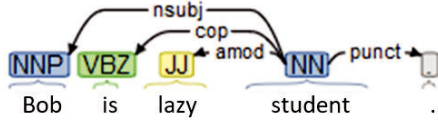


Fig. 9. Assign of type property of property.

Binary relation $((o_l)\iota)\tau\omega$ - presents an extended type that talks about a relationship between two individuals. To determine this type, there should be a dependency *nsubj* or *nsubjpass*, where dependent component is individual and the independent component is a relation. Then dependency *dobj* with another individual as the dependent component and relation as the independent component should be found (shown in Figure 10).

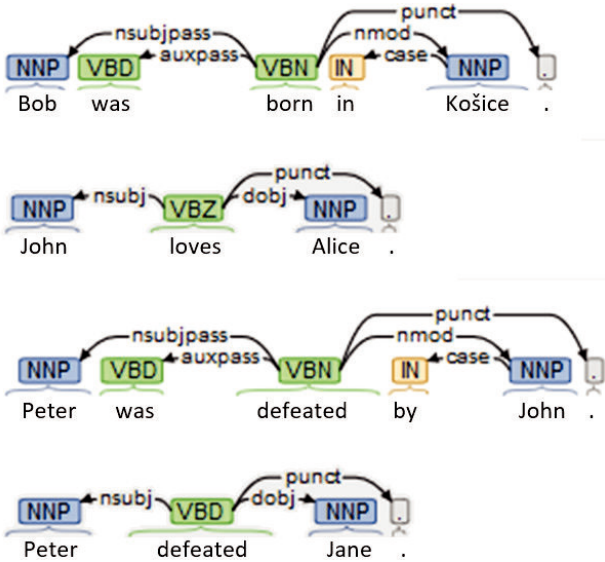


Fig. 10. Assign of binary relation type.

Value τ - presents an extended type. An expression has this type when it is a cardinal number *CD*. Value can be expressed by the special nouns as *weight, height, length, temperature, depth, width, distance, speed, count, number...* (shown in Figure 11).

Property of the value $((\tau)\tau)\omega$ - presents an extended type related to all kinds of value. After identifying the value type in the sentence, the dependency *nsubj* should be found, where the dependent component is the value and the independent component is the property (shown in Figure 11, fourth sentence).

All of the above-mentioned types, which are used in the first phase of natural language analysis, have been implemented in our TIL semantic machine. Correct type analysis is a prerequisite for the second phase of semantic analysis - synthesis of the construction, which is discussed in the next chapter.

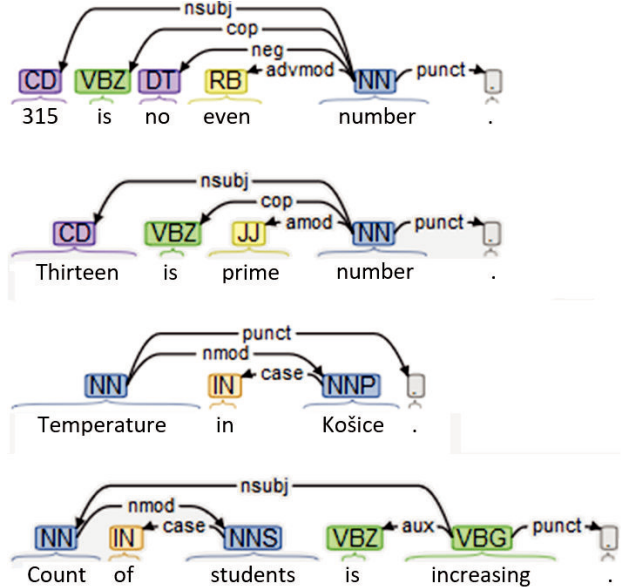


Fig. 11. Assign of value type.

B. Creation of constructions

As is shown in Figure 3 in json object, which is returned as the output from Stanford CoreNLP, is a dependency *ROOT*. The advantage of this dependency is that it is in every analyzed sentence, so the creation of construction always starts from it. When the type analysis is done, a list of all found types is created. Scanning of types that were assigned during the type analysis is needed to determine a type of sentence *ROOT*.

According to sentence *ROOT* type, it is discovered which other types are related to the sentence *ROOT*. If the sentence „Peter is student.” is analyzed, the *ROOT* of sentence is the word *student* which is the property type. Discovering of related types is necessary because if the word *student* is related to an individual then the word *student* has to be intensionalized for the output construction.

$$\lambda w \lambda t [[[\sim^0 \text{student } w] t] \sim^0 \text{Peter}]$$

After all related types are discovered, then the object is created for every analyzed word. Each object is in a specific relationship to the other objects of the sentence, allowing to construct the sentence construction (through closure and composition). In some cases, more lexical units can represent one type. E.g., in sentence „Count of cars in Kosice is decreasing.” words *count, of, cars* are analyzed by Stanford CoreNLP as independent words, but together they are a value type. In the final construction, they are put together as one object - *count_of_cars*.

$$\lambda w \lambda t [[[\sim^0 \text{decreasing } w] t] \sim^0 \text{count_of_cars_in_Kosice}]$$

Another important step is to identify conjunctions if they are in the sentence. For sentence „Peter’s wife is great teacher and awesome mother.” found *ROOT* is *teacher*. After related types

are discovered, the algorithm determines if *ROOT* contains conjunction by finding dependency *conj*.

$$\lambda w \lambda t \left[\left[\left[\left[\text{teacher } w \right] t \right] \left[\text{great } w \right] t \right] \right] \lambda w \lambda t \left[\text{Peter} \left[\left[\text{wife } w \right] t \right] \right] \wedge \left[\left[\left[\text{mother } w \right] t \right] \left[\left[\text{awesome } w \right] t \right] \right] \lambda w \lambda t \left[\text{Peter} \left[\left[\text{wife } w \right] t \right] \right] \right]$$

At last, it must be checked if there is a negation in the sentence. For sentence „Matthew was not born in Kosice.‛, the following output is generated:

$$\lambda w \lambda t \left[\neg \left[\left[\text{born } w \right] t \right] \text{Matthew } \text{Kosice} \right]$$

Previous constructions show outputs generated from TIL semantic machine. The presented implementation is functional and useful for further research related to the detection of semantic ambiguities of natural language expressions.

IV. CONCLUSION

A functional prototype of TIL semantic machine is introduced in the paper. This unique implementation is only the second in the world which is based on TIL principles and the first-ever which allows to process English sentences and create TIL constructions from them.

Even though our development of the semantic machine was successfully finished, here are many possibilities for how it would be desirable to extend the semantic machine. From Figure 4 it is obvious that the last (voluntary) step of the TIL sentence analysis - type control has not been implemented. The usefulness of this step is in the implementation of a fully automated correctness check of the created constructions.

It would also be interesting to implement a solution that includes a proper syntactic analyzer which can replace Stanford CoreNLP. The last interesting extension is the creation of a universal semantic machine that can create TIL constructions from sentences in several world languages.

V. ACKNOWLEDGEMENT

This work was supported by the following projects:

- Faculty of Electrical Engineering and Informatics, Technical University of Košice under the contract No. FEI-2018-59: Semantic Machine of Source-Oriented Transparent Intensional Logic.
- Slovak Research and Development Agency under the contract No. SK-AT-2017-0012: Semantics technologies for computer science education.

REFERENCES

- [1] P. Tichý, “The Foundations of Frege’s Logic,” in *De Gruyter, Berlin and New York*, 1988.
- [2] J. Raclavský, “On Partiality and Tichý’s Transparent Intensional Logic,” *Magyar Filozófiai Szemle*, vol. 54, pp. 120–128, 2010.
- [3] A. Horák, “The Normal Translation Algorithm in Transparent Intensional Logic for Czech,” Ph.D. dissertation, Faculty of Informatics, Masaryk University, Brno, 11 2001.
- [4] M. Duží and P. Materna, “Logical Form,” *G. Sica, Essays on the Foundations of Mathematics ad Logic*, pp. 115–153, 2005.
- [5] J. Perhác, D. Mihályi, and L. Maťaš, “Resource Oriented BDI Architecture for IDS,” in *Informatics 2017*, 2017, pp. 293–299.
- [6] J. Perhác, D. Mihályi, and V. Novitzká, “Modeling Synchronization Problems: From Composed Petri Nets to Provable Linear Sequents,” in *Acta Polytechnica Hungarica*, vol. 14, 2017, pp. 165–182.
- [7] V. Novitzká, W. Steingartner, and J. Perhác, “From Coalgebraic Logic to Modal Logic: An Introduction,” in *IPSI BgD Transactions on Internet Research. - Belgrade (Serbia)*, vol. 15, 2019, pp. 39–44.
- [8] R. Montague, “The Proper Treatment of Quantification in Ordinary English,” in *Approaches to Natural Language*, 1970.
- [9] L. T. F. Gamut, *Logic, Language, and Meaning, Volume 2: Intensional Logic and Logical Grammar*, University of Chicago Press, 1990.
- [10] M. Duží and P. Materna, *TIL jako procedurální logika, Průvodce zvidavého čtenáře Transparentní intensionální logikou*. Slovak Republic: aleph Bratislava, 2012.
- [11] D. Sleator, “Link Grammar,” <https://www.link.cs.cmu.edu/link/>, Accessed: 8.5.2019.
- [12] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: <http://www.aclweb.org/anthology/P/P14/P14-5010>
- [13] M. C. de Marneffe and C. D. Manning. (2008, Sep.) Stanford Typed Dependencies Manual.
- [14] M. Duží, “Til as the Logic of Communication in a Multi-Agent System,” *Research in Computing Science, special issue Advances in Natural Language Processing and Applications*, vol. 3, pp. 27–40, 2008.

A case study on the effects and limitations of refactoring

Bianka Flóra Békefi

Eötvös Loránd University, Budapest, Hungary
Email: bbianka@inf.elte.hu

Kristóf Szabados

Eötvös Loránd University, Budapest, Hungary
Email: Kristof.Szabados@ericsson.com

Attila Kovács

Eötvös Loránd University, Budapest, Hungary
Email: Attila.Kovacs@inf.elte.hu

Abstract—We present design issues and empirical observations on extending an industrial development environment with a new refactoring tool for *Feature Envy* code smells. Refactoring is generally considered to improve code quality, however, it seems from multiple studies that such tools may introduce new bugs and are rarely used by the developers. We tried to discover through a case study why they are not used more often and how they could be made more useful in large scale development.

We implemented a refactoring feature executing *Move Method* refactoring on TTCN-3 projects and performed a survey with industrial experts working in large scale software development projects.

In our empirical research we found that the refactoring method preferred grants more control to the users. By doing so, additional factors can be considered while refactoring besides metrics, such as human factors and company policies.

Index Terms—TTCN-3, refactoring, code smells, Feature Envy

I. INTRODUCTION

During the development of a software, multiple different factors can lead to the deprecation of quality. Factors, such as changes of the members of the team, problems due to the growth of the software, and in the long term, faults or problems that are noticed only thanks to the newer and better tools. Furthermore, with time the structure of a software system might diverge from the original design. With each new requirement supported, the quality of the code can slightly decrease, implementing new features, fixing bugs become more and more difficult and time-consuming. At the same time, new software design patterns could be discovered and quickly evolving programming languages might offer better constructs, ways of working.

Such events can make the developers realize that the software architecture is no longer ideal, it could be organized in a better way. As the number of code smells increases (indicating a deeper problem with the architecture), the need for improving the structure of the program without changing its behaviour emerges. But this is not easy on large scale software. When not addressed previously, large scale systems can create large amount of code smells, to the point where even for easily fixable issues, their amount might make the task seem unfeasible. In such large systems, tool support for refactoring should be a minimum as tracking dependencies

and side effects of a change can become humanly impossible. However, refactoring tools have to be automated to be efficient in practice. Yet, an overly automated tool might prove to be useless or even doing damage if it does not take into account the people developing the product and the political aspects in the company.

Through a case study and a survey, we have developed a refactoring to identify and remove *Feature Envy* code smell in TTCN-3 codes and analyzed the feedback from the users, we observed its effect on code quality and the attitude of the developers towards using refactoring tools. Our contribution is unique, as we placed a large emphasis on the feedback and on how it would be possible to make more useful refactoring tools.

During the development of the refactoring tool for TTCN-3, we made observations that can be applied for other programming languages too, as these findings do not depend on the language or the IDE, but come from the organizational structure of the company that develops the software.

The paper is organized as follows. In Section II we present some necessary terms and the previous works related to motivations behind refactoring and its impact on code quality. Section III describes the code smell and the proposed solution to correcting them through which we executed our research and made observations. The results of our implemented algorithm is shown in Section IV followed by the feedback received in V. Finally, Section VI deals with the validity of our results, Section VII shows our summary and Section VIII offers ideas for further research.

The paper is based on research results achieved by a collaboration between Ericsson Ltd. and the Large Scale Testing Research Lab of Eötvös Loránd University, Budapest.

II. BACKGROUND AND PREVIOUS WORKS

A. Code smells

Martin Fowler first introduced the concept of code smells in object-oriented programs in 1999 [1]. In his definition, code smells in the code structure might indicate a deeper problem in the software. This problem is not necessarily a bug in the functionality, it can be a design flaw which increases the probability of bugs in the future.

However, only indicating code smells is not enough, on large systems the cost of correction might be prohibitively expensive and removing them manually can cause even greater damage. As developers don't like spending time on removing code smells [8] and managers financing weeks or months of work that provides no obvious immediate benefit for their customers, providing automated refactorings seems to be the necessary.

B. Refactoring

According to a representative definition by Fowler et al., "Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure." [1]

One goal of refactoring can be identifying code smells and removing them while preserving the functionality of the system. By improving the structure of the software, the code can be more easily understood and its complexity might decrease, thus both maintaining and changing the software can require less time.

C. Motivations behind refactoring

It was shown in [8] that only a third of the developers try to avoid anti-patterns and even less (2%) use tools to identify and remove such behaviors. Furthermore, the rate is much lower among testers (12% and 1%). Another study [9] found that even if code smell detecting and refactoring tools are available, developers do not refactor the code more often. However, if they are motivated or their attention is drawn to the importance of code quality, their productivity can be increased by using such tools. A study on code smells in object-oriented code [10] has similar findings, as according to their paper more than half of the code smells persist through different versions of the software and the removal of the rest was rarely due to refactoring.

It seems that the academic and industry definition of refactoring differ. Kim et al. [11] observed, that in contrast to the definition of refactoring by Fowler [1], 78% of the developers define it in a way that refactoring improves some aspects of the behavior, e.g. performance. The developers asked said, that they do most of the refactorings manually, even when they had automated refactoring tools available. ~77% of the developers thought that refactorings might introduce new bugs. The main motivations behind refactoring were found to be poor readability, poor maintainability, code duplication and difficulties with repurposing the code.

Paloma et al. [12] found that developers generally do not consider some code smells, such as *Middle Man*, *Long Parameter List*, *Lazy Class* and *Inappropriate Intimacy*, as problems that need to be refactored. Those that are considered more important and concerning were *Complex Class*, *Large Class*, *Long Method* and *Spaghetti Code*.

According to Silva et al. [13], more than half of the developers refactor manually, and only 38% of them use tools provided by an IDE to refactor due to distrust in the tool, or that the changes needed were so simple that using a

tool was unnecessary. Another study [14] found that 90% of the refactorings are done manually, without tools. Black and Murphy-Hill [15] suggest that the lack of using refactoring tools is due to the fact that developers prefer floss refactoring¹ over root canal refactoring², however, most of the tools are created for the second strategy.

D. Impact of refactoring on code quality

Contrary to the popular belief that refactoring improves code quality, studies show that it can have the opposite effect. Weigerber and Diehl [16] found that refactorings might increase the occurrence of bugs in the code. In contrast, Moser et al. [17] found indicators that refactoring leads to less complex code, simpler design and increased maintainability of the software.

Ferreira et al. [18] observed that refactorings often introduce new bugs. However, changes made during root-canal refactoring are less prone to bugs than those made during floss refactoring.

Shatnawi and Li [19] measured the effects of refactoring by four software quality metrics and observed that certain refactorings improved software quality while others deteriorated it. According to their findings, about half of the refactorings improved, around 19% deteriorated, and one third of them did not change the software quality.

Cedrim et al. [20] observed that 80% of the refactorings were performed on code smells. However, only 9.7% removed code smells, while 57% did not remove them and 33% even introduced new ones that were not removed afterwards. According to another study [21], only 42% of the refactorings are performed on code smells and as low as 7% of them remove code smells.

E. TTCN-3

TTCN-3 [2]³ is a programming language developed by ETSI⁴ specifically for testing and certification.

The fast development of the programming languages might raise the need for refactoring. Similarly to other programming languages, such as C++ [3] and Java [4], new TTCN-3 editions are released more often recently, they are continually maintained and improved. Practices that were perfectly fine in a previous version might turn out to be suboptimal in the new version. The programs sometimes should be refactored to take advantage of the new improvements to create a maintainable, comprehensible, well-structured software.

III. CASE STUDY

The chosen code smell and refactoring to use for our research were *Feature Envy*[1] and the related *Move Method*

¹The process when developers refactor the code while simultaneously changing the behavior as well, such as adding a new feature or fixing a bug.

²Refactoring, which is done with the sole purpose of improving the code structure.

³Testing and Test Control Notation Version 3

⁴European Telecommunication Standards Institute

refactoring. The proposed tool we implemented detects *Feature Envy* code smells in code written in TTCN-3 and recommends possible solutions through *Move Method*.

In the case of TTCN-3 we defined it in two different ways: *Feature Envy* based on accessed data and based on components⁵. In the former case, a function in a module does not use any features from its own module but contains references to features from other modules. In the latter case, we assume that there are several functions that run on the same component but they might not be in the same module.

The algorithm is divided into three main parts. First, *Feature Envy* code smells are detected in the modules. Then, all possible target modules are computed, and the optimal one is chosen based on different metrics or criteria. Finally, the methods are moved into the selected module and the missing dependencies are inserted.

The main difference of our method compared to most semi-automated ones [22] is that we give more control to the users to revise the suggested locations and decide whether they find them appropriate. It was necessary as certain aspects, such as the design and the architecture of the project, cannot be easily measured through metrics. At the first step, the users can select which modules to analyze from the project. Afterwards, they can decide which ones from the functions that are recommended to be relocated they want to move. And lastly, they can choose the destination finding method and can filter the possible destinations by module names.

A. Detect methods to move

Detecting which methods should be moved from its module is based on the fields and methods the given function uses. Each statement is examined in the function's body to see if those refer to another method, field or other feature from some module. Only features that are defined outside the inspected method but defined within the module are considered. If the function does not use anything from its own module and it is not private, it should be moved elsewhere.

B. Find destinations for Move Method

After having collected the methods to be moved, the possible target modules have to be identified.

During the refactoring, the users can choose between 3 different methods to find the destination module.

1) *Choosing the shortest module*: One of the approaches chosen was to choose the shortest module, thereby evening out the length of the modules in order to avoid creating *God Classes*. As Cedrim et al. [20] observed, *Move Method* refactoring increased the occurrence of *God Class* code smell by 35% and this option could possibly avoid that. The drawback of choosing the shortest module is that the dependencies are not taken into consideration, leading to an unchecked explosion in the architecture's complexity.

⁵Components are execution entities on which test cases or functions are executed. They are defined as types and become active during execution. A component represents a computer, thus with multiple components a distributed system can be simulated. The behavior of a component is defined by the functions that run on it.

2) *Choosing a destination that results in the least new dependencies*: Another approach was choosing the destination based on the number of new dependencies introduced. The goal was to minimize this value, and, as a result, keep the complexity increase minimal. However, the drawback of this method is that it can result in creating *God Classes* because moving everything into the same module would be the best way to reduce the number of dependencies.

3) *Choose destination based on components*: The third way of finding a new destination is based on the component the function runs on. The goal is to move the function to the module which contains the most of the methods running on the same component. This way, the methods that belong together, as if they were methods of a class, will be located in the same module.

In all the three cases, the possible new locations are presented in a list and rated between 0 and 100 as a percentage, indicating to what extent that module is recommended.

C. Insert missing dependencies

After the final destination for the method has been selected, the dependencies have to be updated.

There are two different kind of updates that might have to be done. One of them is at the new location of the function to make sure that none of its necessary dependencies is missing, so that the module can be compiled.

The other one is in those modules that referenced the original location of the function. They have to be checked to see if they contain references to the function and if they do, whether the new location is present among the dependencies or it has to be added.

Another modification that might be made, although it is not necessary for the software to compile, is to remove unused dependencies that refer to the previous location of the method. The refactoring tool presented in this paper does not remove them automatically, however, another tool is available to organize the imports and remove unused ones.

IV. RESULTS

There are multiple software quality metrics available for TTCN-3 codes in the Eclipse Titan [27] plug-in. Those that reflect the changes best caused by the implemented refactoring are lines of code, afferent and efferent coupling, number of imports and external feature envy. We measured the effect of the refactorings on two projects to see if they improve those values and remove *Feature Envy* code smells. However, they were performed without any background knowledge of the projects.

For this reason, during measuring the refactoring's effects we always chose the best recommendation except when the name of the module suggested that it contains only type definitions. In that case, if there was such a possible destination whose name did not suggest that it contained only types, this module was chosen. If there was no such destination, the function was left in its original module. In both projects, a large module was chosen to be refactored whose name did

	External feature envy	Afferent coupling	Efferent coupling	Imports
Before refactoring	255540	60305	51933	7993
Shortest module	255166	60100	51925	8183
Minimal new imports	254846	60079	51855	8137
Components	254593	59965	51797	8169

TABLE I
RESULTS OF THE ALGORITHMS ON THE INDUSTRY PROJECT.

	External feature envy	Afferent coupling	Efferent coupling	Imports
Before refactoring	8074	3932	3081	527
Shortest module	7742	3774	3037	552
Minimal new imports	7960	3885	3033	536
Components	8070	3931	3100	552

TABLE II
RESULTS OF THE ALGORITHMS ON THE 3GPP_IMS TEST SUITE.

not suggest that it served to group together functions or types and contained movable functions.

The tables below show the results of the algorithms measured on two projects; Table I on an industry project [9] and Table II on the 3GPP_IMS [5] test suite.

All three refactoring approaches improved the software quality metrics. The values decreased after the refactoring except the number of imports, and the efferent coupling in the case of relocating the functions based on the components. However, deciding which should be used in a given situation depends on the architecture of the system.

V. USER FEEDBACK

In this section we show the feedback the users gave during our research and at the end of the project. To make sure our research is useful in practice, we tried to get industry experts involved during development, and collect feedback afterwards.

A. Feedback during development

During the development of the tool, it was tested by future users who gave feedback on the need for further features, presence of bugs or changes needed to be made.

Originally, two approaches were implemented, finding the shortest module and the one with the minimal new dependencies needed. At first, only the best destination was displayed, the potential destination modules could not have been filtered by their name and the whole selected project was processed. The users found that, although the algorithm worked well, it was hard to respect the design of the product and the policies of the company.

As some modules were created by another team of developers or for different products, and the refactoring should not be made across these different fields. Different teams can be located in different countries with distinct parts of the code they are responsible for. Moving code would also mean transferring responsibility, which would lift the scope of the refactoring to management level (hiring, firing, reorganization, etc.). Thus, they requested that the subjects of the refactoring could be limited in a way, that only those modules would

be refactored that are selected, not the whole project. This way, both the modules from which methods are moved and the destinations would be among the selected ones. As they know to which team the modules and libraries belong, they can avoid unpleasant situations that might otherwise come up.

Another aspect to consider was the semantics of the projects. The code was often organized in a way, that the types and functions were grouped together in different modules with distinct names that referred to their contents (with a postfix of `_types`, `_functions`, etc.). Although nowadays it isn't considered an ideal structure, in older projects it has to be respected. To do so during refactoring, we added a filtering option for the possible destination names. This way, when used, the modules with such names will not be considered as a candidate for new location.

As the users wanted more control over the algorithm, instead of automatically moving the function to the best location, the users can choose a different module than the proposed one. The possible new destinations are listed, each of them with a value that shows to what extent that location is recommended.

Also, they requested another destination finding approach, finding the new location based on the component the method runs on. The components can be imagined in a way as if they were classes turned inside out. Those functions that run on the same component can be considered as if they belonged to a class. The main difference is, however, that they do not have to be in the same module.

Originally, the refactoring automatically executed the following steps:

- 1) Select the methods from the modules
- 2) Choose the destination for each method
- 3) Display final results (compare the original locations to the new ones).

As a result of the feedback, the steps have been rearranged the following way to give more control to the users:

- 1) List the movable functions from the selected modules
- 2) The users choose the functions to be moved from the list
- 3) List the destinations for the methods and the filtering options
- 4) The users choose the desired destinations from the list
- 5) Display final results (compare the original locations to the new ones).

B. Feedback at the end of the project

At the end of the project, we asked 5 experts to try the tool and answer the following questions to measure their satisfaction.

1) *How useful are those refactorings that are executed on the whole project (even if they can deal with only a part of the occurring problems)?*: 75% of the people asked considered them useful, however, such refactorings often cannot be performed as team borders have to be respected. Features that belong to one team cannot be moved to a file for which development and maintenance responsibility belong to another one.

2) *How useful is the automatic relocation of the functions?:* More than half of the respondents found it useful (with the limitations mentioned in the previous question), especially for modules or projects which haven't been carefully planned.

3) *Is it useful that in the first step the user can filter the modules on which to execute the refactorings?:* They all found it to be great help, both for selecting only those modules that belong to that team. One of them found it "very useful to build trust in the tool on a smaller set of modules."

4) *How good is it that at the last step the tool recommends multiple destinations (even if there is a best one)?:* More than three quarters of the answers were "really useful" or "quite helpful". In some cases the tool might not recognize aspects that influence the destination but the user can take them into account.

5) *How disturbing is it when a refactoring that analyzes a whole project takes long time to finish?:* It was not found to be a problem at all. According to them it is "not an issue, if it is an otherwise efficient algorithm" and "not disturbing as long as a weekend is enough".

6) *Which is the most useful among the three approaches in selecting the destination module? Is any of them useless?:* 40% of the people found the approach based on the used components the most useful, while 60% considered the one which inserted the lowest number of new dependencies.

7) *How convenient is rating the possible new locations with percentage?:* Although generally they thought it was convenient, a possible improvement they mentioned was to tell the users how the rating is calculated.

VI. THREATS TO VALIDITY

The research was conducted on the TTCN-3 programming language, so there might be limits to generalizing the results. However, as not only the source code was taken into consideration, the results might be valid in case of other languages as well.

Although we managed to ask only five developers for their feedback, this was roughly the available number of system architects at the company in that field.

As the projects on which we measured our tool were either open source or belonged to the same industry partner, the results might be affected by the policy of the company or the culture of the country. In companies with very different structure, different results might be found. In smaller projects the company policy very likely will not cause any troubles.

VII. SUMMARY

According to the measured values, all three refactoring approaches improved the software quality metrics. However, deciding which option should be used in a given situation depends on the architecture of the system.

We found that company policy and management aspects, team borders and system limitations play such a big role in the changes that can be made to a project, that in many cases, although refactoring would be useful in improving the quality of the code and the system's structure, it can not be executed.

This seems to be in accordance with Conway's law [28], which says that an organization's communication structure is mirrored in the systems' structure they design. The human factor plays such a big role in refactoring, that it should be considered when a tool is created.

These observations can be applied for other programming languages too, for example large object oriented projects. There might arise such a situation, that, although the refactoring would respect the structure of the code and the OOP principles (or any other programming principles depending on the language), it cannot or should not be executed due to organizational structure. Such an example is moving a method from the derived class to the base class; this refactoring seems sensible and is often supported by different IDEs. However, if the two classes are developed by two different departments or teams, it is likely that such a refactoring cannot be executed.

Independently of the programming language, organizational structure can have a large influence on refactoring. Our findings suggest that, especially in case of large projects, users prefer having more control over how the refactoring is done. By finding the right balance between creating automated refactoring tools and asking the users for input even if it isn't essential, refactoring tools might be made more useful.

VIII. FURTHER WORK

There are still several possibilities to increase the usability and effectiveness of the refactoring tool.

Besides the three methods available, further approaches could be implemented. One possible addition is moving the function to the module where it is used the most, which contains the highest number of references to the function.

Furthermore, the effects of human factors and human design on the code quality and refactoring could be researched further.

REFERENCES

- [1] Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, *Refactoring: Improving the Design of Existing Code*, 2002.
- [2] ETSI: Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language.
- [3] <https://en.cppreference.com/w/cpp/language/history>
- [4] <https://www.codejava.net/java-se/java-se-versions-history>
- [5] <http://www.ttcn-3.org/index.php/downloads/publics/publics-3gpp/77-3gpp-ims-test-suite>
- [6] <http://www.ttcn-3.org/index.php/downloads/publics/publics-etsi/23-publics-dpmr>
- [7] <http://www.ttcn-3.org/index.php/downloads/publics/publics-etsi/25-publics-ipv6>
- [8] Attila Kovács, Kristóf Szabados *Knowledge and mindset in software development - how developers, testers, technical writers differ - a survey* in: E. Vatai (ed.): Proceedings of the 11th Joint Conference on Mathematics and Computer Science, Eger, Hungary, 20th - 22nd of May, 2016.
- [9] Kristóf Szabados and Attila Kovács, *Internal quality evolution of a large test system industrial study* in Acta Universitatis Sapientiae, Informatica, 2016.
- [10] Alexander Chatzigeorgiou and Anastasios Manakos: Investigating the Evolution of Bad Smells in Object-Oriented Code. Innovations in Systems and Software Engineering. 10. 106 - 115. 10.1109/QUATIC.2010.16.
- [11] Miryung Kim, Thomas Zimmermann, Nachiappan Nagappan, *A Field Study of Refactoring Challenges and Benefits* in IEEE Transactions on Software Engineering, 2014.

- [12] Fabio Palomba, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, Andrea De Lucia, *Do they Really Smell Bad? A Study on Developers Perception of Bad Code Smells* in Software Maintenance and Evolution (ICSME), 2014.
- [13] Danilo Silva, Nikolaos Tsantalis, Marco Tulio Valente, *Why We Refactor? Confessions of GitHub Contributors* in Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016). ACM, New York, NY, USA, 858-870.
DOI:10.1145/2950290.2950305
- [14] Emerson Murphy-Hill, Chris Parnin, Andrew P. Black, *How We Refactor, and How We Know It* in IEEE Transactions on Software Engineering, 2012.
- [15] Andrew P. Black, Emerson Murphy-Hill, *Why Dont People Use Refactoring Tools?*. Computer Science Faculty Publications and Presentations. Paper 115, 2007.
- [16] Peter Weigerber, Stephan Diehl, *Are Refactorings Less Error-prone Than Other Changes?* Conference Paper, Proceedings of the 2006 International Workshop on Mining Software Repositories, 2006.
- [17] Raimund Moser, Pekka Abrahamsson, Witold Pedrycz, Alberto Sillitti, Giancarlo Succi, *A Case Study on the Impact of Refactoring on Quality and Productivity in an Agile Team*. Conference Paper, 2007.
- [18] Isabella Ferreira, Eduardo Fernandes, Diego Cedrim, Anderson Ucha, Ana Carla Bibiano, Alessandro Garcia, Joo Lucas Correia, Filipe Santos, Gabriel Nunes, Caio Barbosa, Balduino Fonseca, Rafael de Mello, *Poster: The Buggy Side of Code Refactoring: Understanding the Relationship between Refactorings and Bugs* in ICSE 18 Companion: 40th International Conference on Software Engineering Companion, May 27- June 3, 2018.
- [19] Raed Shatnawi, Wei Li, *An Empirical Assessment of Refactoring Impact on Software Quality Using a Hierarchical Quality Model* in International Journal of Software Engineering and its Applications, 2011.
- [20] Diego Cedrim, Alessandro Garcia, Melina Mongioli, Rohit Gheyi, Leonardo Sousa, Rafael de Mello, Balduino Fonseca, Mrcio Ribeiro, and Alexander Chavez, *Understanding the impact of refactoring on smells: a longitudinal study of 23 software projects* in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). ACM, New York, NY, USA, 465-475.
DOI: 10.1145/3106237.3106259
- [21] Gabriele Bavota, Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, Fabio Palomba, *An Experimental Investigation on the Innate Relationship between Quality and Refactoring* in Journal of Systems and Software, May 2015.
- [22] Stephan Erb, *A survey of software refactoring tools*. Technical report, 2010.
- [23] Marios Fokaefs, Nikolaos Tsantalis, Alexander Chatzigeorgiou, *JDeodorant: Identification and Removal of Feature Envy Bad Smells*. Conference Paper, 23rd IEEE International Conference on Software Maintenance and Evolution, ICSM 2007.
- [24] Vitor Sales, Ricardo Terra, Luis Fernando Miranda, Marco Tulio Valente, *Recommending Move Method Refactorings using Dependency Sets* in Reverse Engineering (WCRE), 2013.
- [25] Christian Napoli, Giuseppe Pappalardo, and Emiliano Tramontana, *Using Modularity Metrics to Assist Move Method Refactoring of Large Systems*, 2013, in Proceedings of the 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS '13). IEEE Computer Society, Washington, DC, USA, 529-534.
DOI: 10.1109/CISIS.2013.96
- [26] Rocco Oliveto, Malcom Gethers, Gabriele Bavota, Denys Poshyvanyk, Andrea De Lucia, *Identifying method friendships to remove the feature envy bad smell: NIER track*. Conference Paper, 33rd International Conference on Software Engineering, 2011.
- [27] Titan, <https://projects.eclipse.org/projects/tools.titan>, last visited: January, 2019.
- [28] Melvin E. Conway, *How do committees invent?* in Datamation, 1968.

Synergy of Organizational Patterns and Aspect-Oriented Programming

1st Peter Berta

*Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Slovakia
pepoberta@gmail.com*

2nd Valentino Vranić

*Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Slovakia
vranic@stuba.sk*

Abstract—By observing Conway’s law, which explains how organizing people directly affects the code they produce, and by taking into account the specifics of different programming paradigms, a lot can be done to improve the software development process. We analyzed organizational patterns of agile software development from the perspective of aspect-oriented programming, whose aim is to improve the separation of concerns in code, which, in turn, allows for better separation of tasks performed by people. We find aspect-oriented programming to be highly related to at least nine out of more than a hundred of organizational patterns in Coplien and Harrison’s catalog, namely: Work Split, Team per Task, Sacrifice One Person, Divide and Conquer, Conway’s Law, Form Follows Function, Shaping Circulation Realms, and Hallway Chatter. Aspect-oriented programming can support organizational patterns in division of labor, treating distractions, and increasing decoupling between the software modules developed by different parts of the organization. On the other hand, organizational patterns can help mitigate pointcut fragility and aspect obliviousness in general.

Index Terms—modularization, asymmetric aspect-oriented programming, symmetric aspect-oriented programming, organizational patterns, Conway’s law, use cases, people, agile software development

I. INTRODUCTION

Conway’s law reveals a direct connection between people and code. It states that organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations [1]. Consequently, an organization that develops a software system should be structured the way that this software system is desired to be structured.

To structure means to separate components and interlink them in an organized way. This is driven by a human need to act separately. Programming paradigms have their limitations in what can be separated. For example, in Java, a method can’t be defined outside a class and then made a part of it. But using an aspect-oriented extension to Java called AspectJ, this can be achieved.

Good ways of structuring software development organizations have been documented as organizational patterns [2]. As with other software patterns and patterns in general, they deal with balancing contradicting forces. For example, (software) architects need to focus on the overall structure, but they should not lose contact with the development reality. The Architect also Implements pattern puts these forces into a balance by letting the architect participate in actual programming [2].

In this paper, we analyze what organizational patterns are particularly suitable for aspect-oriented programming and vice versa. For this, we apply scientific reasoning to identify analogies between aspect-oriented programming techniques and organizational patterns. This analysis might be useful to those who would like to apply aspect-oriented programming and good organizational practices along. Section II briefly explains aspect-oriented programming. Section III explains how we approached identifying suitable organizational patterns. Section IV presents the project management patterns and their relationship to aspect-oriented programming. Section V presents the organizational style patterns and their relationship to aspect-oriented programming. Section VI discusses related work. Section VII concludes the paper.

II. ASPECT-ORIENTED PROGRAMMING

At the peak of the popularity of aspect-oriented programming, there were dozens if not hundreds of aspect-oriented programming languages available. Given the nature of aspect-oriented programming, they mostly represented aspect-oriented extensions to established programming languages. Consequently, there were many variants and classifications of aspect-oriented programming. Among these, a particularly important distinction is whether aspects are perceived as special modules that affect the base modules or the whole design is built out of aspects. The former is known as asymmetric aspect-oriented programming, while the latter is being denoted as symmetric aspect-oriented programming [3].

Although originally related only to HyperJ [4], an abandoned IBM’s prototype aspect-oriented programming language, symmetric aspect-oriented programming features can be observed in programming languages not explicitly denoted as aspect-oriented, such as Scala (traits), Ruby (open classes), and JavaScript (prototypes) [5]. Furthermore, symmetric aspect-oriented programming can be emulated in AspectJ [5], the reference asymmetric aspect-oriented programming language. It is worth noting that peer use cases (use cases with no dependencies between them) essentially represent symmetric aspect-oriented modularization [5]. Consider the use case diagram depicted in Figure 1. Place an Order and Cancel an Order are two peer use cases. As can be seen in Figure 2, the realization of these use cases is based around the same classes, but seen differently from the perspective—or aspect—of each use case.

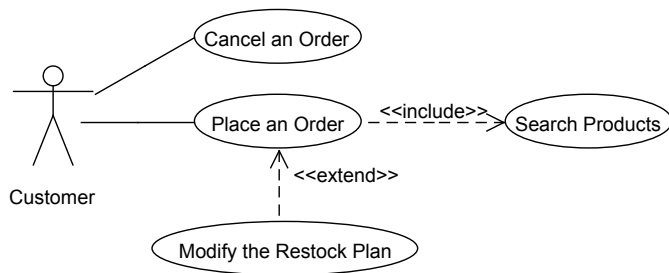


Fig. 1. Use cases as aspects.

The extend relationship represents an instance of asymmetric aspect-oriented modularization, as observed by Jacobson [6], [7]. In the use case diagram depicted in Figure 2, the Modify the Restock Plan use case affects the Place an Order use case with the latter not being aware of this. Contrast this to the include relationship, in which Place an Order explicitly invokes Search Products. This is more evident in the actual use cases:

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products is being activated.
3. Customer confirms the product selection and adjusts its quantity.
4. If the product is available, System includes it in the order.
5. Customer continues in ordering further products.
6. Customer chooses the payment method, enters the payment data, and confirms the order.
7. Customer can cancel ordering at any time.
8. The use case ends.

Extension points:

Checking Product Availability: Step 4

UC Modify the Restock Plan

Alternate Flow: Modify the Restock Plan

After the Checking Product Availability extension point in the Place an Order use case:

1. System checks the available quantity of the product being ordered.
2. If the quantity is below the limit, System adds the

quantity under demand to the restock plan.

3. The flow continues with the step that follows the triggering extension point.

In AspectJ, the extend relationship can be preserved by the following aspect, which enforces restocking the plan instead of having to call this functionality explicitly from the Ordering class:

```

public aspect RestockPlan {
    ...
    void around(Product product):
        call(* Ordering.productAvailable(..) &&
            args(product) {
                ... // increase the product quantity in the restock plan
            }
        ...
}
  
```

Speaking more generally about AspectJ, which is the most elaborated representative of asymmetric aspect programming, we may say that aspects modify program execution in join points, such as method call or execution, constructor call or execution, or field (attribute) access. Join points are determined by pointcuts, which can be seen as declaratively specified sets of join points. Modifications are expressed as so-called advice. A piece of advice is the code executed before, after, or around (instead of) a join point. Aspects can add new elements and inheritance relationships to classes using so-called inter-type declarations, which is a way to emulate symmetric aspect-oriented programming in AspectJ [5].

III. IDENTIFYING ORGANIZATIONAL PATTERNS RELATED TO ASPECT-ORIENTED PROGRAMMING

In identifying organizational patterns suitable for aspect-oriented programming, we focused on Coplien and Harrison’s comprehensive catalog [2], which includes over a hundred patterns forming four interlinked pattern languages. This catalog might be denoted as a culmination of the work that can be traced back to at least 1994 [8]–[10]. Furthermore, Sutherland, Coplien, et al. rephrased Scrum into organizational patterns [11]. All the while, Ambler talks about process patterns [12], and some other authors took the route of warning about bad organizational practices expressing them as antipatterns [13]–[15] (a unifying antipattern catalog has been reported [16]).

In the following two sections, we present the organizational patterns from Coplien and Harrison’s catalog we found to be related to aspect-oriented programming. However, we do not claim there are no other organizational pattern from this catalog or in general to be related to aspect-oriented programming.

IV. PROJECT MANAGEMENT PATTERNS

In the Project Management pattern language, we identified four patterns related to aspect-oriented programming: Work Split, Team Per Task, Sacrifice One Person, and Surrogate Customer. Figure 3 depicts their position within this pattern language. The Size the Schedule pattern is introduced as the

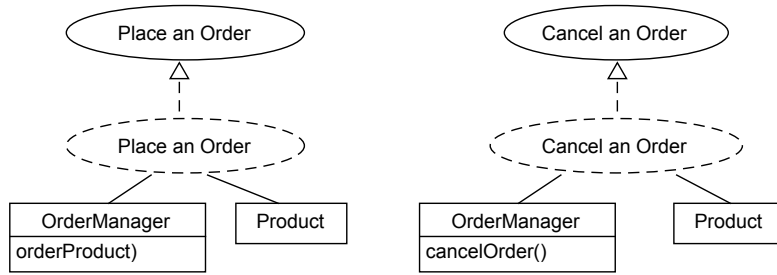


Fig. 2. Realization of peer use cases in the symmetric aspect-oriented way.

root of the Project Management pattern language despite it wasn't among the patterns we identified as being related to aspect-oriented programming. The edges depict the order of the application as proposed by Coplien and Harrison [2]. Dashed edges mean that the order is indirect, i.e., there are other patterns in between.

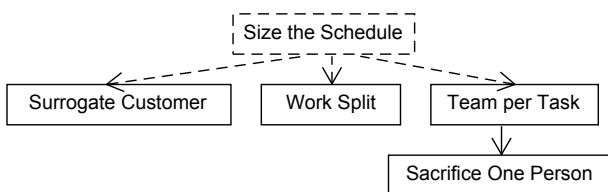


Fig. 3. The patterns related to aspect-oriented programming within the Project Management pattern language.

Each of the following sections treats one organizational pattern. It first brings its brief description based on Coplien and Harrison's catalog [2]. Subsequently, the relationship to aspect-oriented programming is explained.

A. Work Split

During the software development process, big tasks can make it seem that no progress is being made. To make the progress visible, the team should split the task at hand into two parts: the urgent part and deferred part. By focusing on the urgent part, the tasks will get finished and progress will be made, relieving the team from the feeling of being stuck. This is the Work Split pattern.

However, once the team gets onto the deferred part, it may become necessary to adapt the urgent part. Consider a complex use case with several alternative flows or a use case extended by one or more use cases. Obviously, the basic flow of this use case would be an urgent part, while alternative flows and extension use cases would be deferred. In effect, alternative flows are internal extensions, so in both cases asymmetric aspect-oriented programming can be used as presented in Section II.

B. Team per Task

A crisis occurring during the software development process needs to be handled. To ensure that progress will be made and issues solved, a subteam is created to solve the crisis, allowing

the main team to keep working on the main line. This is the Team per Task pattern.

Dividing large development teams into smaller, dedicated teams, when done correctly, is beneficial for the entire software development process. Developers within a dedicated team can fully focus on the task assigned to them. They do not need to draw their attention to other problems that are not theirs to solve. To fully exploit the benefits of such division of labor, the dedicated teams can utilize aspect-oriented programming to keep the implementation of the tasks less coupled. Consequently, they will interfere with each other to a lesser extent and will be able to make progress independently.

C. Sacrifice One Person

The Sacrifice One Person pattern resolves a similar situation as the Team per Task pattern (Section IV-B). The main team can work on the assigned task, while one person is working on several minor distractions. This one person can handle the distractions using aspect-oriented programming to keep the main team free of having to adapt their implementation in order to incorporate the implementation of the distractions. Instead, with asymmetric aspect-oriented programming, the code resulting from the distractions can affect the corresponding join points in the code resulting from the main task.

D. Surrogate Customer

When a software development team needs to make a decision about certain requirements, and the customer is not available, a surrogate customer role may be created. It is assigned to one member of the software development team, who acts and thinks like a customer. This is the Surrogate Customer pattern.

Surrogate Customer role resembles the Cuckoo's Egg aspect-oriented design pattern [17]. With the Cuckoo's Egg pattern, a given object can be replaced by another object, which usually exhibits a (slightly) different behavior. The surrogate customer role can be seen as this new object that is replacing the old object, which, in turn, corresponds to the missing or dysfunctional customer. This is as if aspect-oriented design is applied to organizing people.

V. ORGANIZATIONAL STYLE PATTERNS

In the Organizational Style pattern language, we identified five more patterns related to aspect-oriented programming:

Divide And Conquer, Conway’s Law, Form Follows Function, Hallway Chatter, and Shaping Circulation Realms. Figure 4 depicts their position within this pattern language. As with the Project Management language treated in the previous section, the root of the Organizational Style pattern language, the Few Roles pattern, is introduced despite it wasn’t among the patterns we identified as being related to aspect-oriented programming. Again, the edges depict the order of the application as proposed by Coplien and Harrison [2], while dashed edges mean that the order is indirect, i.e., there are other patterns in between.

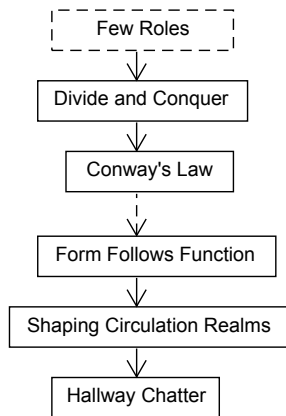


Fig. 4. The patterns related to aspect-oriented programming within the Organizational Style pattern language.

As with the Project Management pattern language, each of the following sections treats one organizational pattern. It first brings its brief description based on Coplien and Harrison’s catalog [2]. Subsequently, the relationship to aspect-oriented programming is explained.

A. Divide and Conquer

Large organizations are hard to maintain and lead. Assigning tasks or communicating requirements among sizable teams is difficult. To make it easier, people should be divided into teams based on their roles. This is the Divide and Conquer pattern.

Dividing people into independent teams and assigning them software modules to be developed directly may be easier with symmetric aspect-oriented programming. This is appropriate for dealing with peer use cases (recall Section II). As with the Team per Task pattern (recall Section IV-B), the teams will interfere with each other to a lesser extent and will be able to make progress independently.

B. Conway’s Law

As has been said in the introduction, Conway’s law states that organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations [1]. Consequently, an organization that develops a software system should be structured the way that this software system is desired to be structured. This can

be perceived as an organizational pattern: the Conway’s Law pattern.

This is a general pattern and, in general, aspect-oriented programming can help in achieving a greater level of decoupling between the software modules developed by different parts of the organization. This includes reversing dependencies if necessary: recall the implementation of the extend relationship between use cases presented in Section II).

C. Form Follows Function

The Form Follows Function pattern recommends grouping together roles that are similar, manipulate similar artifacts, or operate within the same domain.

One of the main goals of aspect-oriented programming is to untangle source code and make it more comprehensible. Form Follows Function also focuses on grouping similar roles and creating a system from disarranged structure. By creating groups with similar activities and employing symmetric aspect-oriented programming similarly to the implementation of peer use cases mentioned in Section II, code gains certain level of structure or modularity, and the development process becomes easier to maintain.

D. Shaping Circulation Realms

Communication is crucial for organizations. While this is not specific to software development, the collaborative nature of software development tends to make the importance of communication more visible. A manager cannot expect communication to be spontaneous, but rather should encourage it by creating structures that make it effortless. This is the Shaping Circulation Realms pattern. By establishing communication structures, it serves as a basis for other patterns.

Pointcut fragility is a long-known issue in aspect-oriented programming [18]. It is caused by the changes in identifier names and by the context in which they occur, such as moving a method to a different class or a class to a different package [18]. For example, this pointcut captures all the calls to methods provided by the Stock class whose name begins with calc:

```
pointcut calculations(): call(* Stock.calc*(..));
```

Calls to the methods such as calculateRevenues() or calculateDifference(). The pointcut is robust enough to capture the calls to the calculating methods regardless of their argument list. It would even survive a slight change in the naming policy such as that calculating method names begin with calc or cal. However, if this becomes compute, the pointcut would break, i.e., fail to capture what it was intended to.

Existing strategies to decreasing pointcut fragility focus on improving the way they are expressed by, for example, using sufficiently general regular expressions rather than specific identifier names or relying on annotations to methods and fields (attributes) rather than on their names. However, much of pointcut fragility can be avoided by encouraging communication among developers, which is where Shaping Circulation Realms can help. More organized communication will help

developers make more stable decisions regarding identifier names on one side and pointcut definitions on the other side, as well as being informed of inevitable changes early enough to take appropriate actions.

E. Hallway Chatter

All the people on a project can't communicate all the time, but all the people need to be informed what's happening on a project in a timely manner. This means that there must be some way of spreading news and even gossip throughout different parts of the organization, which typically reside in different parts of the building. The news should be allowed to spread in informal communication among some of the people belonging to these different parts of the organization. This is the Hallway Chatter pattern.

Aspect-oriented modularization is based on affected code being oblivious of the aspects that affect it. However, this means that the developers or teams whose code is affected by the aspects others have developed may be oblivious of them, too. Consequently, they may wrongfully interpret the changes in the program behavior and lose a lot of time until they identify the real source of the problem. This is particularly prominent with otherwise unannounced changes introduced in dire straits. Hallway Chatter can contribute to raising the awareness of what's happening even with respect to such seemingly very low level programming decisions at least by just spreading the gossip that some aspects are being prepared to get over the current situation.

VI. RELATED WORK

Others have tried to relate organizational and development perspectives, too. Thus, Muller [19] reports that aspect-oriented programming is beneficial to agile software development iterations in terms of both time they take and code that needs to be developed. This speaks in favor of our findings how aspect-oriented programming is in line with some organizational patterns of *agile* software development. However, Muller's study is limited to AspectJ, hence ignoring symmetric aspect-oriented programming.

Parsons [20] managed to unify the aspect-oriented software development ontology published by van den Berg et al. [21] with his own agile software development ontology, which also speaks in favor of our findings. Parsons also observes difficulties in identifying the relationships between some notions.

Santosa et al. [22] report on how pair programming, which Coplien and Harrison consider to be an organizational pattern [2], is related to aspect-oriented programming related mistakes. Pair programming seems to increase mistakes in implementation logic, choice of the advice type, and code duplication, while decreasing mistakes in compilation, refactoring completeness, and excessive refactoring.

Burrows et al. [23] report a study on performing maintenance tasks with aspect-oriented programming in a pair programming constellation. The study showed certain implementation strategies to be more fault-prone than others such as

specific techniques for accessing data from base code modules and binding advice to pointcuts.

Apart from Coplien and Harrison [2], throughout whose organizational patterns of agile software development resonate general software development practices, Sagenschneider [24] reports a pattern language for aligning object-oriented and general programming practices with organizing people in an office. This ranges from using comments to categorize work to using multithreading to get more tasks done.

Pícha et al. [25]–[27] propose a common model for project pattern analysis. They focus on detection of organizational antipatterns in software artifacts, such as repository commits, but not in code itself.

VII. CONCLUSIONS AND FURTHER WORK

By observing Conway's law, which explains how organizing people directly affects the code they produce, and by taking into account the specifics of different programming paradigms, a lot can be done to improve the software development process. We analyzed organizational patterns of agile software development from the perspective of aspect-oriented programming, whose aim is to improve the separation of concerns in code, which, in turn, allows for better separation of tasks performed by people.

We find aspect-oriented programming to be highly related to at least nine out of more than a hundred of organizational patterns in Coplien and Harrison's catalog [2], namely: Work Split, Team per Task, Sacrifice One Person, Divide and Conquer, Conway's Law, Form Follows Function, Hallway Chatter, and Shaping Circulation Realms. Aspect-oriented programming can support organizational patterns in division of labor, treating distractions, and increasing decoupling between the software modules developed by different parts of the organization. On the other hand, organizational patterns can help mitigate pointcut fragility and aspect obliviousness in general.

Next steps should lead to the patterns that the ones we analyzed refer to, i.e., to considering pattern sequences or pattern sublanguages [28]. We assume that the most promising candidates are those patterns referred to by several other patterns proven to be related to aspect-oriented programming, such as: Organization Follows Location (5), Gate Keeper (4), Organization Follows Market (4), and Engage Customers (3). The number in parentheses indicates how many patterns out of those proven to be related to aspect-oriented programming refer to the given pattern.

The People and Code pattern language embraces some patterns that are very close to design patterns, such as Standards Linking Locations, Variation Behind Interfaces, Hierarchy of Factories, Parser Builder, Factory Method, and Loose Interfaces, whose aspect-oriented implementation should be explored for its relatedness to effective people organization.

It would be interesting to explore how are organizational patterns related to microservices as a popular architectural style, in particular for their noted uses in combination with aspect-oriented programming [29].

ACKNOWLEDGMENTS

The work reported here was supported by the Scientific Grant Agency of Slovak Republic (VEGA) under the grant No. VG 1/0759/19 and by the Research & Development Operational Programme Research and Innovation for the project Research of advanced methods of intelligent information processing, ITMS NFP313010T570, co-funded by the European Regional Development Fund.

REFERENCES

- [1] M. E. Conway, “How do committees invent?” *Datamation*, vol. 14, no. 4, pp. 28–31, 1968.
- [2] J. O. Coplien and N. B. Harrison, *Organizational Patterns of Agile Software Development*. Prentice-Hall, 2004.
- [3] W. H. Harrison, H. L. Ossher, and P. L. Tarr, “Asymmetrically vs. symmetrically organized paradigms for software composition,” IBM Research, Tech. Rep. RC22685, 2002.
- [4] H. Ossher and P. Tarr, “Multi-dimensional separation of concerns and the hyperspace approach,” in *Software Architectures and Component Technology*. Kluwer, 2002.
- [5] J. Bálík and V. Vranić, “Symmetric aspect-orientation: Some practical consequences,” in *Proceedings of NEMARA 2012: International Workshop on Next Generation Modularity Approaches for Requirements and Architecture, AOSD 2012*. Potsdam, Germany: ACM, 2012.
- [6] I. Jacobson and P.-W. Ng, *Aspect-Oriented Software Development with Use Cases*. Addison-Wesley, 2004.
- [7] I. Jacobson, “Use cases and aspects – working seamlessly together,” *Journal of Object Technology*, vol. 2, no. 4, 2003.
- [8] J. O. Coplien, “Borland software craftsmanship: A new look at process, quality and productivity,” in *Proceedings of 5th Borland International Conference*, Orlando, FL, USA, 1994.
- [9] J. O. Coplien and J. Erickson, “Examining the software development process,” *Dr. Dobbs’ Journal of Software Tools*, vol. 19, no. 11, pp. 88–95, 1994.
- [10] J. O. Coplien, “A generative development-process pattern language,” in *Pattern Languages of Program Design*, J. O. Coplien and D. C. Schmidt, Eds. ACM Press/Addison-Wesley Publishing, 1995, pp. 183–237.
- [11] J. Sutherland, J. O. Coplien *et al.*, *A Scrum Book: The Spirit of the Game*. The Pragmatic Bookshelf, 2019, <https://sites.google.com/a/scrumplp.org/published-patterns/book-outline> – <http://www.scrumbook.org/>.
- [12] S. W. Ambler, *Process Patterns: Building Large-Scale Systems Using Object Technology*. Cambridge University Press, 1998.
- [13] W. J. Brown, H. W. S. M. III, and S. W. Thomas, *AntiPatterns in Project Management*. John Wiley & Sons, 2000.
- [14] W. J. Brown, R. C. Malveau, H. W. M. III, and T. J. Mowbray, *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*. John Wiley & Sons, 1998.
- [15] P. A. Laplante and C. J. Neill, *Antipatterns: Identification, Refactoring, and Management*. Auerbach Publications, 2005.
- [16] P. Brada and P. Pícha, “Software process anti-patterns catalogue,” in *Proceedings of the 24th European Conference on Pattern Languages of Programs, EuroPLoP ’19*. Irsee, Germany: ACM, 2019, to appear.
- [17] R. Miles, *AspectJ Cookbook*. O’Reilly, 2004.
- [18] C. Koppen and M. Stoerzer, “PCDiff: Attacking the fragile pointcut problem,” in *Proceedings of 1st European Interactive Workshop on Aspects in Software, EIWAS 2004*, Berlin, Germany, 2004.
- [19] J. Muller, “What are the benefits of aspect oriented programming to project iterations developed using agile processes?” 2005, <https://pdfs.semanticscholar.org/d04e/d891fb64ea396f58689137971455727e2b51.pdf>.
- [20] D. Parsons, “An ontology of agile aspect oriented software development,” *Research Letters in the Information and Mathematical Sciences*, vol. 15, pp. 1–11, 2011.
- [21] K. van den Berg, J. M. Conejero, and R. Chitchyan, “AOSD ontology 1.0 – public ontology of aspect-orientation,” AOSD-Europe, Tech. Rep. IST-2-004349-NOE AOSD-Europe, 2005, https://www.researchgate.net/publication/232905610_AOSD_Ontology_10_-_Public_Ontology_of_Aspect-Orientation.
- [22] A. Santosa, P. Alvesa, E. Figueiredoa, and F. Ferrari, “Avoiding code pitfalls in aspect-oriented programming,” *Science of Computer Programming*, vol. 119, no. C, pp. 31–50, 2016.
- [23] R. Burrows, F. Taïani, A. Garcia, and F. C. Ferrari, “Reasoning about faults in aspect-oriented programs: A metrics-based evaluation,” in *Proceedings of 2011 IEEE 19th International Conference on Program Comprehension, ICPC 2011*. Kingston, ON, Canada: IEEE, 2011.
- [24] D. Sagenschneider, “OfficeFloor: Using office patterns to improve software design,” in *Proceedings of 18th European Conference on Pattern Languages of Programs, EuroPLoP 2013*. Irsee, Germany: ACM, 2015.
- [25] P. Pícha and P. Brada, “ALM tool data usage in software process metamodeling,” in *42nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2016*. Limassol, Cyprus: IEEE, 2016.
- [26] P. Pícha, P. Brada, R. Ramsauer, and W. Mauerer, “Towards architect’s activity detection through a common model for project pattern analysis,” in *Proceedings of 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017*. Gothenburg, Sweden: IEEE, 2017.
- [27] P. Pícha and P. Brada, “Software process anti-pattern detection in project data,” in *Proceedings of the 24th European Conference on Pattern Languages of Programs, EuroPLoP ’19*. Irsee, Germany: ACM, 2019, to appear.
- [28] W. Sulaiman Khail and V. Vranić, “Treating pattern sublanguages as patterns with an application to organizational patterns,” in *Proceedings of 22nd European Conference on Pattern Languages of Programs, EuroPLoP ’17*. Irsee, Germany: ACM, 2017.
- [29] T. Cerny, “Aspect-oriented challenges in system integration with microservices, SOA and IoT,” *Enterprise Information Systems*, vol. 13, no. 4, pp. 467–489, 2018.

Virtual Reality Test Setup for Visual Impairment Studies

Ingmar Besic

Faculty of Electrical Engineering
University of Sarajevo
Sarajevo, Bosnia and Herzegovina
ingmar.besic@etf.unsa.ba

Zikrija Avdagic

Faculty of Electrical Engineering
University of Sarajevo
Sarajevo, Bosnia and Herzegovina
zikrija.avdagic@etf.unsa.ba

Kerim Hodzic

Faculty of Electrical Engineering
University of Sarajevo
Sarajevo, Bosnia and Herzegovina
kerim.hodzic@etf.unsa.ba

Abstract—Visual impairment severely constraints the ability to independently conduct many everyday tasks that we usually do not consider challenging. Although some types of visual impairment can be treated efficiently there is still a considerable number of visually impaired persons, especially among aging population, which depend on help of others or assistive technology to sustain their life quality. Visually impaired person cannot perceive the full extent of surrounding information due to the lack of visual details. However great progress can be achieved if surrounding information can be somehow visually transformed to the subset of visual information that visually impaired person can perceive. To certain extent every impairment has individual characteristics, as different individuals may better respond to different subsets of visual information. Thus any assistive solution aiming to visually transform surrounding information to accommodate broad range of impairment conditions must be personalized in order to be effective. Virtual reality enables individuals to experience imaginary surroundings by tricking their visual senses and such virtual surroundings can be personalized to any extent desired. We use virtual reality, image processing, and RFID to create a test setup able to simulate visual impairment and visually transformed surroundings suitable for visual Impairment studies. The test setup enables gathering information and conducting studies on both healthy and visually impaired individuals in a controlled environment enabling reliable assistive technology development and testing.

Keywords—Virtual Reality, Visual Impairment, Image Processing, RFID

I. INTRODUCTION

Statistics show that economic burden of visual impairments is enormous. It is estimated that annual economic burden of vision loss, eye diseases and vision disorders in the U.S. only is \$139 billion [1]. This burden might become even higher in the future due to the aging population in developed countries. Major eye diseases are age-related macular degeneration (AMD), glaucoma, diabetic retinopathy, cataract and refractive error. In the U.S. cataract is by far the most frequent eye disease, followed by diabetic retinopathy, glaucoma and AMD, as can be seen in Fig. 1.

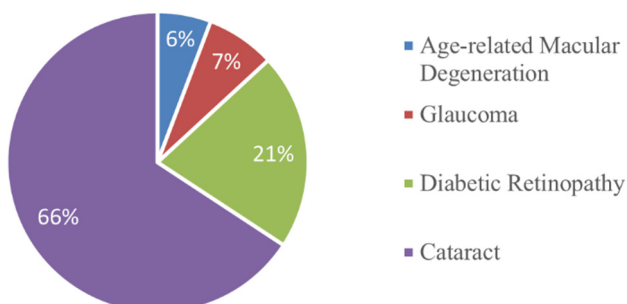


Fig. 1. Major eye disease statistics, North America (U.S.).

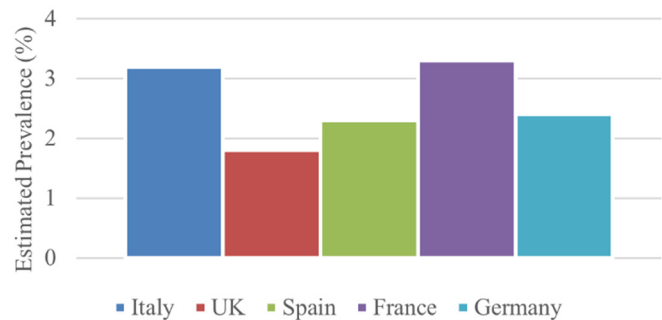


Fig. 2. Late age-related macular degeneration prevalence in Europe.

Some eye conditions are more severe than others. AMD along with diabetic eye disease (DED) is the main cause of blindness and severe visual loss in Europe [2]. One in four Europeans over the age of 60 is affected by AMD. Percentage of affected people varies from country to country and late AMD seems to be the most prevalent in Italy and France, as can be seen in Fig. 2. It is estimated that AMD currently affects around 34 million people in the European Union. The number of patients affected by AMD in the EU is expected to rise by almost 25% based on population growth and ageing until 2050 [2]. Eye disease treatment statistics from 2014 show that African countries, like Botswana, have very similar distribution of eye diseases to the U.S., once treatments of a less severe conditions like conjunctivitis, refractive errors, hordeolum and chalazion are excluded from the statistics [3]. Again, among major eye diseases cataract is by far the most dominant, as can be seen in Fig. 3.

Statistics indeed show that eye disease differences exist among countries, but it seems that these difference cannot be contributed to any specific socio-economic factor. Further, it seems that major eye diseases are rather equally spread over the world population and it is certain that all countries pay significant economic price due to the eye diseases burden. There is also a strong global evidence that despite significant efforts in terms of prevention of blindness, significant growth in the number of older people experiencing sight loss is likely to be experience [4].

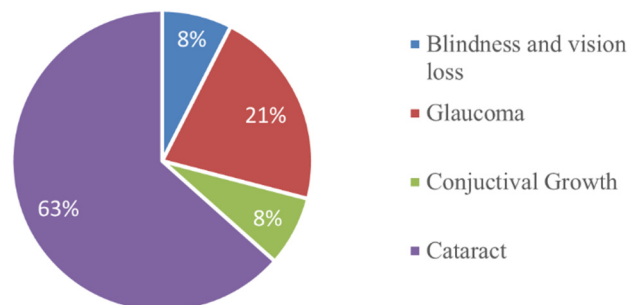


Fig. 3. Major eye disease statistics, Africa (Botswana).

In this paper we use virtual reality headset, image processing, GPU video encoding and RFID indoors localization to create a test setup able to simulate visual impairment and visually transformed surroundings suitable for visual Impairment studies. The test setup enables gathering information and conducting studies on both healthy and visually impaired individuals in a controlled environment enabling reliable assistive technology development and testing.

II. VISUAL IMPAIRMENT SETUP REQUIREMENTS

In order to create suitable test environment that enables reliable assistive technology development and testing, it is important to understand the effects of various visual impairments. It is also important to compare the benefits of existing treatments to the use of assistive technology and specifically to estimate if virtual reality (VR) based assistive technology would be a reasonable choice.

- Age-related macular degeneration causes central vision loss as can be seen in Fig. 4 top-right. There is no cure for AMD, but some treatments may delay its progression or even improve vision [5]. Visually impaired person suffering from AMD would benefit from the use of assistive technology. VR headset based assistive technology could be used to manipulate field of view (FoV) by transferring some visual information from the damaged central FoV to the peripheral FoV. The result would be limited because visual information would be lost while mapping full FoV to a smaller area. More importantly, visual information would be lost because peripheral vision cannot process visual information at the same level of details as the central vision.
- Glaucoma initially causes peripheral vision loss, as can be seen in Fig. 4 middle-left. If not treated it eventually results in blindness. Glaucoma can be treated with

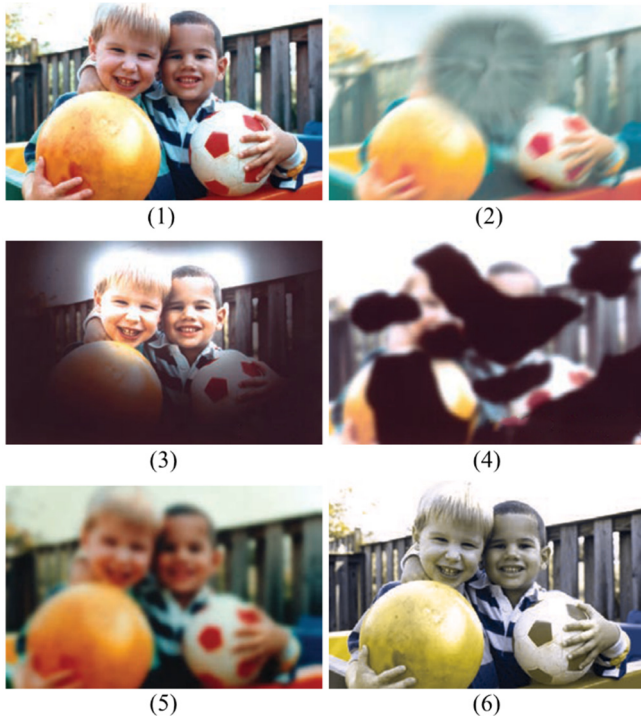


Fig. 4. Visual impairment effects: (1) normal vision, (2) age-related macular degeneration, (3) glaucoma, (4) diabetic retinopathy, (5) cataract, (6) color vision deficiency.

medicated eye drops, micro-surgery and laser treatments [5]. Unfortunately, glaucoma has no early warning signs and is often diagnosed once the optical nerve has been damaged and the lost vision cannot be restored. Visually impaired person with glaucoma induced damage would benefit from the use of assistive technology. VR headset based assistive technology could be used to manipulate field of view (FoV) by transferring some visual information from the peripheral FoV to the central FoV. The result would be somewhat limited because information loss would occur.

- Diabetic retinopathy causes damage both to central and peripheral vision, as can be seen in Fig. 4 middle-right. As diabetes is a lifelong condition, treatment depends on the type of diabetic retinopathy and is aimed at slowing or stopping progression of the condition. Surgery often slows or stops the progression, but it's not a cure [5]. Visually impaired person with diabetic retinopathy induced damage could rely on some forms of assistive technology. However, VR headset based assistive technology would probably not be convenient as damaged areas are dispersed and of irregular shapes.
- Cataract induces blur to both to central and peripheral vision, as can be seen in Fig. 4 bottom-left. It can be treated with a surgery [5]. Visually impaired person suffering from cataract could rely on some forms of assistive technology till the surgery. However, use of VR based assistive technology could not be justified as surgery to remove clouded lens and replace it with a clear, plastic intraocular lens would be more economical and would give much better results.
- Color vision deficiency (CVD) is inability to distinguish certain shades of color, as can be seen in Fig. 4 bottom-right. CVD is inherited condition and cannot be cured [5]. Visually impaired person suffering from CVD would benefit from the use of assistive technology and specifically from VR headset based assistive technology that could be used for a real time image correction aimed at transforming the full FoV image to a CVD friendly image-

Thus, virtual reality is a reasonable choice for assistive technology aimed at visually impaired persons affected with AMD, glaucoma and CVD. In case of AMD, VR based assistive technology is expected to have limited results due to the lack of a central vision. In case of glaucoma VR based assistive technology might have somewhat better results as central vision could be used to compensate peripheral vision damage. And finally, VR based assistive technology have a potential to give the best result if used by visually impaired persons affected by CVD [6]. CVD is caused by malfunction of photoreceptors in human eye retina and there are three classes of cones with photoreceptors susceptible to color: the first class is sensitive to long wavelength in the visible spectrum (L), the second class is sensitive to middle wavelength in the visible spectrum (M) and the third class of cones is sensitive to short wavelengths in the visible spectrum (S). Person with normal color vision has all three cone classes intact and is referred to as trichromat. Malfunction of any of the cone classes will result in CVD.

III. VIRTUAL REALITY TEST SETUP FOR CVD

Virtual reality can be transformed into augmented reality by using optical see-through devices or video see-through devices. Optical see-through device has the advantage of displaying the reality through a clear optics and rendering virtual contents only. However it might introduce mismatches while synchronizing rendered virtual contents with the reality. Video see-through device renders both virtual contents and reality at the same time resulting in possibly higher latency, but with no synchronization mismatches. In the proposed test setup we use video see-through device to avoid synchronization mismatches. Setup is based on a commercial virtual reality headset and stereo camera.

Virtual headset uses PenTile AMOLED display with 2160×1200 resolution (1080×1200 per eye) and 456ppi pixel density [7]. The display is refreshed at the 90Hz rate and seen through hybrid Fresnel lenses enabling 110° field of view. User head motion is tracked using accelerometer, gyroscope and IR LED head tracking using external constellation camera enabling 6 degrees of freedom (6 DoF). Headset supports 3D audio and is connected to a personal computer with 4 meters long HDMI 1.3 and USB 3.0 cables.

Stereo camera is based on two 1/3" backside illumination 4 million pixels image sensors with high low-light sensitivity [8]. The image sensors have synchronized rolling shutter, auto exposure, gain and white balance. The camera has $f/2.0$ aperture lenses that cover field of view 90° (H) x 60° (V) x 110° (D). The camera uses gyroscope and accelerometer to determine its position with +/- 1mm accuracy and orientation with +/-0.1° accuracy. Supported output video modes are: 2.2K mode with 4416x1242 output resolution at 15 frames per second, 1080p mode with 3840x1080 output resolution at 30 frames per second, 720p mode with 2560x720 output resolution at 60 frames per second, and WVGA mode with 1344x376 output resolution at 100 frames per second. In addition to the video, the camera outputs depth range for each pixel. Depth range is between 15 cm and 1200 cm.

Image captured by the stereo camera is Red Green Blue (RGB) coded and this does not correspond to the human photoreceptor's sensitivity. That is why RGB color coding is not suitable for CVD image processing. Captured image must be converted to LMS color space to match L, M and S cones in the retina. Color space conversion is done by the following image processing transformation matrix:

$$\mathbf{M}_{RGB \rightarrow LMS} = \begin{bmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{bmatrix} \quad (1)$$

Transformation from LMS color space to RGB color space is done using inverse $\mathbf{M}_{RGB \rightarrow LMS}$ matrix:

$$\mathbf{M}_{LMS \rightarrow RGB} = (\mathbf{M}_{RGB \rightarrow LMS})^{-1} \quad (2)$$

Thus $\mathbf{M}_{LMS \rightarrow RGB}$ transformation matrix is as follows:

$$\mathbf{M}_{LMS \rightarrow RGB} = \begin{bmatrix} 0.08094 & -0.13050 & 0.11672 \\ -0.01024 & 0.05401 & -0.11361 \\ -0.00036 & -0.00412 & 0.69351 \end{bmatrix} \quad (3)$$

Visually impaired person affected with CVD will see the image transformed into limited color space due to the lack of

certain class of cones. Matrix that simulates CVD condition by transforming image into such limited color space is dependent on the class of cones missing. The most common type of CVD is deuteranopia and it is caused by the lack of M cones. Matrix simulating deuteranopia is as follows [9]:

$$\mathbf{D}_{CVD} = \begin{bmatrix} 1.42319 & -0.88995 & 1.77557 \\ 0.67558 & -0.42203 & 2.82788 \\ 0.00267 & -0.00504 & 0.99914 \end{bmatrix} \quad (4)$$

Thus, visually impaired person affected by deuteranopia will not see the original RGB colors, but rather colors given by the following transformation into a limited CVD color space.

$$\begin{bmatrix} R_{CVD} \\ G_{CVD} \\ B_{CVD} \end{bmatrix} = \mathbf{M}_{LMS \rightarrow RGB} \mathbf{D}_{CVD} \mathbf{M}_{RGB \rightarrow LMS} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5)$$

Image that results from processing transformation (5) can be used to demonstrate how visually impaired person with deuteranopia actually sees the image. Thus virtual reality setup turns a person with normal vision into a deuteranope once the image processing algorithm based on (5) is activated.

IV. VISUAL IMPAIRMENT FRIENDLY RFID ROOM

Both visually impaired and elderly people experience problems related to locating objects of interest. For elderly person it can be challenging to cognitively track object location over the period of time and if object is not in plain sight it can be frustrating experience to locate it. Thus we use RFID technology in our test setup, as can be seen in Fig. 5. Thus the test setup can be used to evaluate RFID object localization and its use by visually impaired persons.

Radio Frequency Identification (RFID) is a mature technology and frequently used in retail, storage, and logistics. At its core, RFID system is composed of a reader and a tag. The basic setup can be expanded to any combination of multiple readers and multiple tags to fit particular functional and spatial requirements. Various object of interest are tagged to enable reader to localize them or determine their orientation or state, as can be seen in Fig. 6.

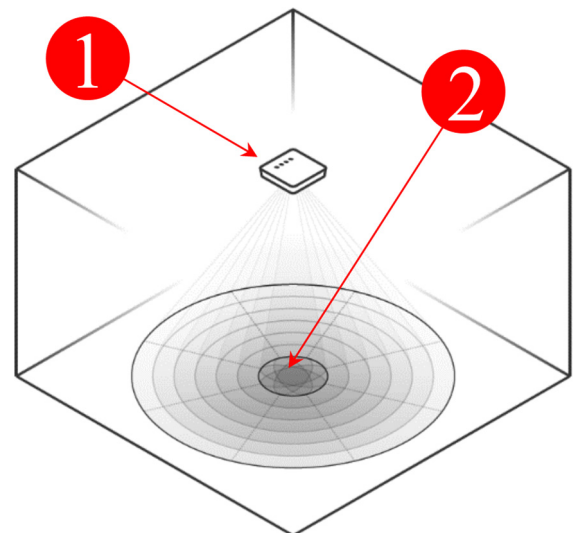


Fig. 5. Visual Impairment Friendly RFID Room: (1) RFID reader at the ceiling, (2) Sector map of the covered region.



Fig. 6. Various objects of interest tagged: (1) Chair with 3 tags attached, (2) desk with 2 tags attached, (3) book with a single tag, (4) desk drawer tagged with 2 tags attached.

A. RFID Tags

The tag is composed of an antenna and an integrated circuit with preprogrammed unique identification code enabling identification to be performed over the wireless communication channel. Tag's integrated circuitry is by several orders of magnitude smaller than antenna, so the RFID tag size depends solely on the antenna size. Having a unique identification code enables tag to virtually represent unique surveyed object or a point of interest at the surveyed object. The major difference between RFID tags and other means of tagging, like bar codes, is the reading speed and reading penetration. RFID tags are superior if batch reading is required, particularly if large number of tags is dispersed over the region of interest with some of the tags out of the line of sight.

From the standpoint of powering the tags, there is a major difference between active and passive tags. Active tag will require an external power source. Although the external source can be any power source with wired connection to the tag, the most common external power source would be an embedded battery. There are certain advantages to using the external power source, like the active tag being continuously operational and thus being able to process external signals, if required and supported by its additional integrated circuitry. However, passive tags have significant advantage over active tags when it comes to the system sustainability. Active tags require embedded battery to be replaced or recharged periodically. Even if active tags power consumption is highly optimized it will still pose a significant obstacle because the number of tags can be significant and it takes time to charge or replace the batteries - leaving the system in void or partially functional state. System based on passive tags will have no issues with sustainability as passive tags are rather affordable and do not have any time limit related to powering or any other longevity issue. The proposed setup is to enable visually impaired individuals to be as much independent as possible. Thus the use of active tags is not the best option as battery recharging and replacement would most likely require external assistance. On the other hand, the use of passive tags results in a sustainable system where external assistance might be required for the initial tagging only. We have evaluated passive tags only as can be seen in Fig. 7.

1) DogBone RFID tag based on Monza R6 integrated circuit features two “dogbone” shaped antennas. Wet inlay variant of the DogBone tag transmits in an omni-directional

pattern eliminating blind spots. Thus the orientation of the tag and reader is less of a factor in the success of wireless communication. Monza R6 integrated circuit provides environmental auto-tune abilities that automatically adjust tag performance for best readability and higher sensitivity. Auto-tune feature can be programmatically turned on and off using low-level commands from the reader. Unlike other tags, the Dogbone integrated circuitry does not have user memory, which provides higher sensitivities and allows the potential for longer read ranges and faster read rates. It also has excellent performance on difficult-to-tag materials.

2) BLING RFID wet inlay tag is based on Monza R6 integrated circuit and designed for jewelry and cosmetics applications. It can be easily converted into small-sized hang-tags for jewelry labels, or attached to small cosmetic items. BLING comes in a tiny 22 mm form factor and is the smallest of all evaluated tags. Its intended use within the test setup is tagging small objects that can be found on the table, like pencils or books, or tagging larger objects that require multiple tags in order to determine their spatial orientation.

3) Belt RFID paper inlay tag is based on Monza R6 integrated circuit and designed for good tolerance and performance on difficult-to-tag or low-detuning materials such as cardboard and plastics and in other demanding environments. It is also a recommended tag for the phased array antenna xArray reader.

4) ShortDipole RFID paper tag is based on Monza R6 integrated circuit and offers excellent performance on lower detuning materials like cardboard, plastics, and corrugated boxes. ShortDipole tag comes in variety of delivery formats – dry, wet, and paper tag – making it suitable for all sorts of labels. In addition to Monza R6 integrated circuit auto-tune abilities that automatically adjust tag performance for best readability and higher sensitivity, ShortDipole tag circuitry supports Enduro technology for better consistency, and Integra technology for data integrity and reliability.

V. EXPERIMENTAL RESULTS

Major goal of the VR test setup is to enable assistive technology design and testing. Assistive device for CVD is supposed to transform the image as seen by the visually impaired person into an image that more closely resembles the original image, but is still bound to the limited CVD color space [6] [7] [8] [9] [14]. In order to do this, image processing must focus on the visual information that visually impaired person misses due to its CVD condition. The missing information arise from the difference between the perceived image and original image:

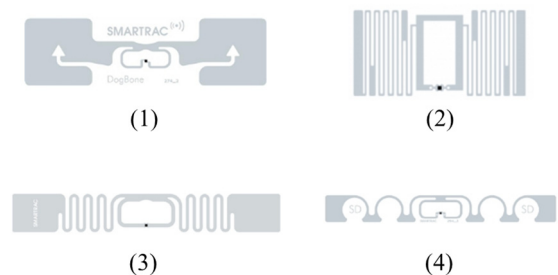


Fig. 7. RFID tags used to evaluate the test setup: DogBone Monza 6 tag (1), BLING Monza 6 tag (2), Belt Monza 6 tag (3) and ShortDipole Monza 6 tag (4).

$$\begin{bmatrix} R_{\Delta CVD} \\ G_{\Delta CVD} \\ B_{\Delta CVD} \end{bmatrix} = \begin{bmatrix} R_{CVD} \\ G_{CVD} \\ B_{CVD} \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6)$$

Virtual reality based assistive device, at its core, performs image processing on the matrix (6) in order to render improved images – the ones that help CVD visually impaired person to better perceive the reality. Thus the VR test setup acts as a prototype assistive device. In order for the prototype to be functional it must not create noticeable delay between stereo camera captured reality images and presented transformed images. Virtual reality requires 90 Hz refresh rate and test setup must be able to achieve this rate with CVD image processing active and running. We have tested the VR setup using known transformation that has achieved positive results while evaluated on 24 visually impaired individuals with CVD condition [9]. The transformation uses following matrix for deuteranopia:

$$\mathbf{D}_T = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} \quad (7)$$

Matrix (7) is applied to the (6) and added to the CVD image (5) that is to be improved:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R_{CVD} \\ G_{CVD} \\ B_{CVD} \end{bmatrix} + \mathbf{D}_T \begin{bmatrix} R_{\Delta CVD} \\ G_{\Delta CVD} \\ B_{\Delta CVD} \end{bmatrix} \quad (8)$$

We have measured refresh rate that can be achieved while CVD image processing is running and results are shown in Fig. 8.

A. Tags Evaluation

X-Stage [10] shown in Fig. 9 is custom designed for tag spatial location evaluation and supports single axis target translation up to 100 mm in 10 micron repeatable steps. X-Stage main components are:

1) Ball Slide Guides: enable smooth and repeatable linear translation of X-axis stage without yaw, pitch and roll [11]. Setup uses two 155 mm long preloaded ball slide guides mounted 70 mm apart to ensure high rigidity. Running parallelism for both vertical and horizontal ball slide guide planes is under 6 microns [10].

2) Anti-Backlash Nuts: enable repeatable translation steps regardless of translation direction changes [12]. Travel direction changes induce small mechanical backlash that must

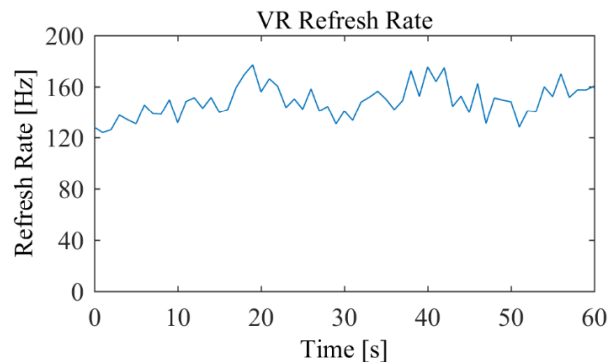


Fig. 8. Virtual reality test setup CVD image processing refresh rate.

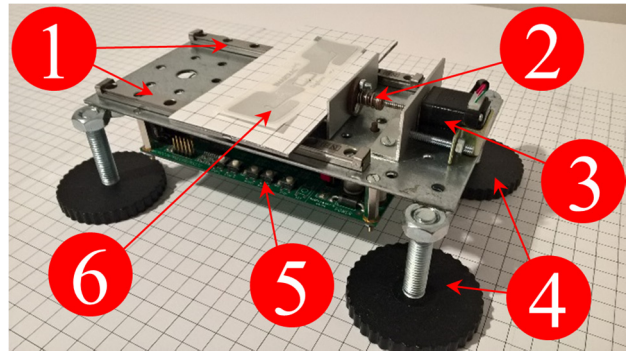


Fig. 9. Custom designed X-Stage for fine tags spatial location evaluation. Ball slide guides (1), anti-backlash nuts (2), linear actuator (3), leveling and support structures (4), micro-stepper controller (5) and tag under the test (6).

be addressed. Anti-backlash nuts use preloaded spring to prevent backlash if axial load on the system is less than spring preload.

3) Linear Actuator: pushes and pulls X-axis stage by converting stepper rotation into anti-backlash nut linear travel. Setup uses hybrid linear actuator with external shaft precisely machined to convert 1.8° rotation steps into 10 micron linear steps. It can move 10 N loads at 1600 steps/second up to 25 N loads at 400 steps/second [13].

4) Leveling and support Structures: enable accurate placing and fastening of components to the optical table or bench (XY plane) and adjusting their height (Z axis) manually. They also serve as vibration isolators [13].

5) Micro-Stepper Controller: controls linear actuator and enables two additional stage axes to be controlled programmatically through software framework, if required. Controller uses two micro-switches as motion limiters and position references. Custom velocity profiles are used to ensure broad range of X-axis stage linear speeds that can be modified in real-time and still avoid stepper vibration and stall [14] [20].

X-Stage is manually placed at desired distance from the reader. Tag under the test is placed on the X-Stage and moved along the preprogrammed path using micro-stepper motion controller connected to the personal computer. Personal computer is connected to the RFID reader over the network. It collects and stores RSSI and AoA readings into the database along with real-time tag position data coming from the micro-stepper motion controller. Stored data can be later processed in order to test and improve accuracy of RFID spatial position estimator algorithms. The major factor to the spatial position estimation is number of tag reads. Spatial position estimation will be more accurate with higher number of tag reads. All evaluated tag types have been tested for number of reads in a limited amount of time and results are shown in Fig. 10.

VI. CONCLUSIONS

In this paper we have proposed a test setup for Visual Impairment Studies. To understand the effects of various visual impairments we have compared the benefits of existing treatments to the use of assistive technology. We have specifically estimated if virtual reality (VR) based assistive technology would be a reasonable choice for certain visual impairments. We affirm that virtual reality is a reasonable choice for assistive technology aimed at visually impaired persons affected with AMD, glaucoma and CVD. In case of

AMD, VR based assistive technology is expected to have limited results due to the lack of a central vision. In case of

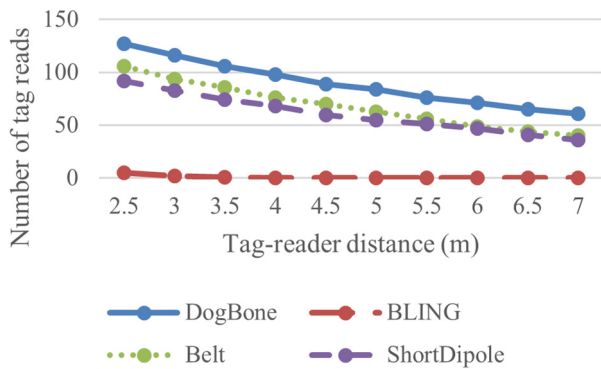


Fig. 10. Number of tag reads vs. tag-reader distance for all evaluated tag types.

glaucoma VR based assistive technology might have somewhat better results as central vision could be used to compensate peripheral vision damage. And finally, we affirm that VR based assistive technology have a potential to give the best results if used by visually impaired persons affected by CVD. We have tested proposed setup as a prototype assistive device for deuteranopia using known color transformation and have achieved real-time performance over required 90 Hz refresh rate for virtual reality

Visual Impairment Friendly Room based on RFID is proposed to enable object localization and its use by visually impaired persons. Various tags types have been evaluated. The DogBone tag proved to have the highest number of tag reads across a range of reader-tag distances and is preferred tag for spatial position estimation.

ACKNOWLEDGMENT

We express our appreciation to the local partners for allowing us to use their NVIDIA GPUs and image capturing equipment for this work.

This work was supported by the Federal Ministry of Education and Science.

REFERENCES

- [1] National Eye Institute, <http://www.nei.nih.gov>. Accessed 2019-08-15.
- [2] European Society of Retina Specialists, <http://www.euretina.org>. Accessed 2019-08-15.
- [3] Statistics Botswana, Eye Disease Statistics 2014, <http://www.statsbots.org.bw>. Accessed 2019-08-15.
- [4] World Blind Union (WBU), <http://www.worldblindunion.org>. Accessed 2019-08-15.
- [5] Mayo Clinic, Diseases & Conditions, <https://www.mayoclinic.org/>. Accessed 2019-08-15.
- [6] I. Besic, S. Omanovic and D. Boskovic, "Time-Domain Color Mapping for Color Vision Deficiency Assistive Technology," *2019 2nd International Conference on Signal Processing and Information Security (ICSPIS)*, DUBAI, United Arab Emirates, 2019.
- [7] Oculus Rift, <https://www.oculus.com>. Accessed 2019-08-15.
- [8] StereoLabs ZED Mini, <https://www.stereolabs.com>, Accessed 2019-08-15.
- [9] P. Melillo *et al.*, "Wearable Improved Vision System for Color Vision Deficiency Correction," in *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 5, pp. 1-7, 2017, Art no. 3800107.
- [10] T. Pramoun, K. Thongkor and T. Amornraksa, "Image watermarking against color blind image correction," *2017 10th International*

Conference on Ubi-media Computing and Workshops (Ubi-Media), Pattaya, 2017, pp. 1-6.

- [6] A. K. Kvittle, H. Oddløyken, P. Green and P. Nussbaum, "Methods for psychophysical assessment of colour difference by observers with a colour vision deficiency," *2018 Colour and Visual Computing Symposium (CVCS)*, Gjøvik, 2018, pp. 1-6.
- [7] J. I. You and K. Park, "Image Processing With Color Compensation Using LCD Display for Color Vision Deficiency," in *Journal of Display Technology*, vol. 12, no. 6, pp. 562-566, June 2016.
- [8] L. Zhang *et al.*, "Improved colour-to-grey method using image segmentation and colour difference model for colour vision deficiency," in *IET Image Processing*, vol. 12, no. 3, pp. 314-319, 3 2018.
- [9] Y. Chen, Chao-Yan Zhou and Long-Yun Li, "Perceiving stroke information from color-blindness images," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 000070-000073.
- [10] I. Besic and Z. Avdagic, "Automated test environment for image processing in laser triangulation 3D scanning," *2016 International Symposium ELMAR, Zadar*, 2016, pp. 161-164.
- [11] W. C. Hsu; C. S. Chang, "Linear ball guide assembly," United States Patent 5,429,439, Jul. 4, 1995.
- [12] K. W. Erikson, K. W. Erikson, "Adjustable preload anti-backlash nut," United States Patent 5,913,940, Jun. 22, 1999.
- [13] I. Besic and Z. Avdagic, "Laser stripe sub-pixel peak detection in real-time 3D scanning using power modulation," *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, 2016, pp. 951-956.
- [14] B. Dwersteg, L. Larsson, "Method and circuit arrangement for sensorless engine load detection and for controlling the motor current in accordance with the load value in stepper motors," United States Patent Application Publication US 2012/0153886 A1, Jun. 21, 2012.
- [20] I. Besic and Z. Avdagic, "Laser stripe model for sub-pixel peak detection in real-time 3D scanning," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 004332-004337.

Analysis of Training Parameters of Feed Forward Neural Networks for WiFi RSSI Modeling

Bence Bogdandy

*Institute of Mathematics and Informatics
Eszterhazy Karoly University
Eger, Hungary
bogdandy.bence@uni-eszterhazy.hu*

Zsolt Toth

*Institute of Mathematics and Informatics
Eszterhazy Karoly University
Eger, Hungary
zsolt.toth@uni-eszterhazy.hu*

Abstract—Modeling WiFi RSSI in buildings is crucial for fingerprinting based indoor positioning algorithms. Fingerprinting is a popular approach which converts the localization task into data mining and optimization problems. Although fingerprinting algorithms may rely on any kind of measurements, WiFi RSSI values are the most commonly used due to their ubiquity. However there are existing solutions for modeling WiFi signal propagation which requires knowledge about the environment and could be difficult to integrate into a positioning algorithm. On the other hand, feed forward artificial neural networks could be used to model the WiFi RSSI values easily based on the fingerprinting database. Therefore, training the networks efficiently is important as is the analysis of training parameters. Experiments over the Miskolc IIS Hybrid IPS Dataset showed that networks with ReLu or tanh activation functions and trained by sgd or adam solvers could achieve high accuracy.

Index Terms—WiFi RSSI, artificial neural network, training

I. INTRODUCTION

Indoor positioning is crucial for providing intelligent location based services in buildings. Outdoor positioning techniques are usually unavailable or unreliable in closed areas due to their unique features, such as signal attenuation and reflection. For example, walls decrease the signal strength so distance estimation based on the path lost is unreliable. Due to these effects, existing indoor positioning methods [8] focus on a wide range of sensors and techniques. Recently the WiFi based solutions gained popularity due to their ubiquity and low installation cost. Fingerprinting [3] is one of the most widely used approach among indoor positioning algorithms that allows the application of data mining techniques and heuristic optimization methods for localization.

Client side positioning algorithms require prior knowledge or model about the environment. Artificial Neural Networks are compact black box models with constant and relatively low time cost. Hence, neural networks could be used on client side to model the environment. Currently, the modeling is focused on the WiFi RSSI values. This paper analyzes the effect of the training parameters for modeling of WiFi RSSI values. The goal is to determine which parameters are the most significant and which settings should be used or avoided.

II. METHODS

Experiments were implemented in Python [7]. Pandas [9] was used to load and transform the data set. Training and

evaluation of the models were performed with the SciKit Learn [10] package.

A. Data Set

Experiments were performed over the Miskolc IIS Hybrid IPS Dataset [13] which is available at the UCI Machine Learning Repository. The data set contains about 1500 measurements which were recorded in a three story building. Figure 1 shows the structure of the data set and it consists of three major parts. The first major part contains the measurement information. Measurements are identified by an UUID which is a 128 bit number. Next, there is the timestamp, when the measurement were performed. The second major part defines the position where the measurement was taken. Absolute position denotes the position in a Cartesian coordinate system. Symbolic position refers to a part of the building and it is denoted by a UUID and a name. The third major part contains the measured values with magnetometer, bluetooth and WiFi. Magnetometer points to the direction of the true north. The following columns denote WiFi access points and Bluetooth devices. If a WiFi access point was available in the location then the corresponding column contains its Received Signal Strength Indication (RSSI) value which is a negative integer. The value Bluetooth devices can be either 0 or 1.

This experiment is focused on the modeling of RSSI of a single access point. Therefore, the Miskolc IIS Hybrid IPS Data Set split into subsets which were used during the training. Each of this sets was limited to a single WiFi access point and the absolute position where the measurement was taken. Then, each data sets was extended with synthetic features which were derived from Cartesian coordinates of the measurement. These synthetic features were the pair-wise product of the coordinates and their representation in spherical coordinate system. The pair-wise product of features is used to learn linear dependency between the features and this synthetic features are denoted by x_y and x_y_z . Spherical coordinates (r, θ, ϕ) are related to the domain of indoor localization and they were introduced because spherical coordinates are used in outdoor localization. For the experiments 32 training sets were generated because there were 32 different access point in the Miskolc IIS Hybrid IPS Data Set. The structure of each data set can be seen in Table I.

Measurement Information		Position Information		Measurements		
Id	Timestamp	Absolute	Symbolic	Magnetometer	WiFi RSSI	Bluetooth
1	2	3–5	6–7	8–10	11 – 42	43 – 65

Fig. 1. Structure of Miskolc IIS Hybrid IPS Dataset [13]

TABLE I
FEATURES OF REGRESSOR TRAINING DATA SET

Column	Description
x	Distance on the X axis of the building
y	Distance on the Y axis of the building
z	Distance on the Z axis of the building
x_y	Calculated feature,
x_y_z	Calculated feature
r	Distance from the Origin, Spherical Coordinate
θ	Azimuthal angle, Spherical Coordinate
ϕ	Polar angle, Spherical Coordinate
target	Measured RSSI value

Modeling WiFi RSSI values is considered a regression problem. The unknown function maps the 8th dimensional real vectors to a single real value. Although there are models [2], [6], [11] for modeling WiFi signal propagation in buildings but these models require information about the floor plan and the characteristics of the access points. Existing fingerprinting data set allows the application machine learning algorithms for modeling the signal propagation. Hence, there is no need for prior knowledge about the floor plan or access points. In addition, feed forward Artificial Neural Networks have been used successfully for regression [12].

B. Training

Multi Layer Perceptron model has been implemented for regression tasks as `MLPRegressor` class of the SciKit Learn Library. `MLPRegressor` was tested with the combination of five different parameter settings which are the activation function, topology, learning rate, alpha and solver. Each neuron in the network use the same activation function which were set to ReLu, tanh or logistic in our experiments. Topology refers to the structure of hidden layers. In the experiments, seven different topologies were tested. Learning rate and alpha were initialized with five different values. Finally, the `sgd`, `lbfgs` and `adam` solvers were used.

Training was performed with the `GridSearchCV` component of the SciKit. `GridSearchCV` performs an exhaustive search over the parameter space and evaluate each combination of settings. This method is used for finding good training parameters which can be tuned later. Cross-validation is used during the training in order to avoid over fitting. Each neural network were evaluated based on its `score` function which calculated the R^2 coefficient of the prediction. Equation 1 shows the calculation of `score` where y denotes the actual values, \hat{y} is the predicted value and \bar{y} refers to the mean of y . Therefore, maximum of the score is 1.0 if there is no difference between the expected and the calculated output. In addition the score can be negative because the model can be arbitrarily worse.

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2} \quad (1)$$

III. RESULTS

Parameter search yielded a data set which contained the training parameters and the achieved score. This data set contains 1575 records with 6 features which are the activation function, alpha, hidden layers, initial learning rate, solver algorithm and achieved score. In addition the data set was extended with calculated features to describe the topology of the network. Finally, the data set was extended with a Boolean feature which decides whether the current network belonged to the top quartile of scores or not.

TABLE II
FEATURES OF PARAMETER SEARCH RESULTS DATA SET

Feature	Type
activation	category
alpha	float64
hidden_layer_sizes	object
learning_rate_init	float64
solver	category
scores	float64
network_size	category
network_avg_neuron_cnt	float64
network_min_neuron_cnt	int64
network_max_neuron_cnt	int64
is_good	bool

Table II shows the features of the result data set. The activation, alpha, hidden_layer_sizes, learning_rate_init and solver represents the corresponding training parameter of the `MLPRegressor`. The activation feature is categorical and its possible values are logistic, relu and tanh. The alpha feature is a real value between 0.0001 and 0.03. Topology of the network is denotes by the hidden_layer_sizes which is an object. In order to facilitate its processing network_size, network_avg_neuron_cnt, network_min_neuron_cnt and network_max_neuron_cnt features were derived. The network_size is the number of layers in the network. The network_avg_neuron_cnt, network_min_neuron_cnt and network_max_neuron_cnt features denotes the average, minimum and maximum number of neurons in the layers. The learning_rate_init feature is the initial learning rate parameter of the training process and it is a real number between 0.0001 and 0.03. The solver feature is categorical and its possible values are adam, bfgs and sgd. The score shows the achieved after training. Finally, the is_good

feature denotes whether the given record belongs to the top quartile by score or not.

A. Score Distribution

The score values show that the trained model achieved 0.7387 score. In the worst case the score was around $-7 \times 10^2 3$. Approximately third of the parameter settings could not achieve a positive score value. On the other hand, half of the settings outperformed 0.2875 score and records in the top quartile have at least 0.6455 score.

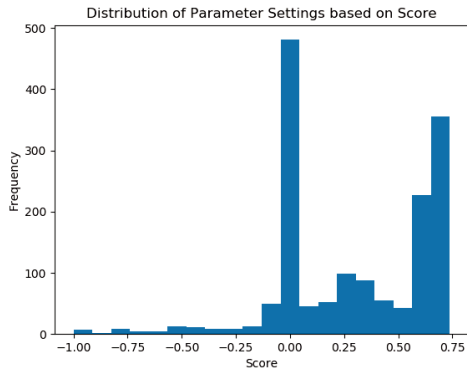


Fig. 2. Score Distribution

Figure 2 shows the histogram of score values. If a score value was less than -1.0 then it was replaced with the lower boundary. Three kind of records can be distinguished in Figure 2. Firstly, there is a peak at 0.0 value and about the third of the records belonged here. Secondly, two groups with the highest score values have more than 200 records. These groups also make up approximately third of the data set. Finally, the last third of the records does not belong to these above two categories. In other words, two third of the records perform good or poorly and the performance of the last third of the data set varies between these two categories.

B. Parameters Significance

Grouping of records could be caused by a single parameter setting. To determine the impact of each feature on the score, the records were classified. The settings were considered good if their score were placed in the upper quantile. This categorization of the records allowed their classification.

Decision Tree [4], [5] was used to classify the settings because it is based on information gain and the tree can be visualized and explained easily. Because Decision Tree building algorithms can operate only with numeric values, the categorical variables (*activation*, *solver*) had to be encoded. Decision tree is built with a top-down approach until the tree reaches the desired depth. It splits the nodes by a feature based on the Gini index. In other words, each node is separated by a feature which produces the biggest and most homogeneous subsets. Visual representation of decision trees are usually used to explain the classification process and explore simple rules.

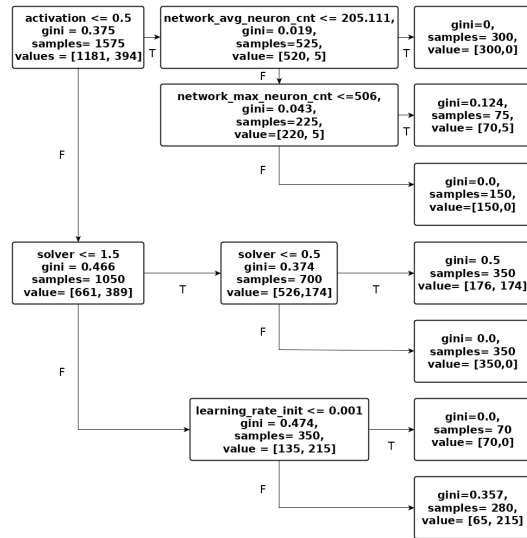


Fig. 3. Decision Tree of Score Performance

Figure 3 shows the classification of the parameters based on their score. The depth of the tree was limited to 3. All the subsets are quasi homogeneous except two leaves. This decision tree is still readable and shows a few important parameter choices.

Firstly, the decision in the root node is made by the activation function feature. This feature is categorical and it was encoded for the decision tree building algorithm. The root condition can be rephrased as whether the logistic activation function was used or not. If logistic activation function is used then 520 settings were considered poor and only 5 of them were good. Due to the low number of good settings, this branch of the tree is not detailed further. The rest of the nodes were not separated by the activation function. So the decision tree advises against the usage of logistic activation function for this task.

Solver was used two times on the other branch of the tree. Because only three solvers were tested, each solver can be distinguished in the tree. On the first level, the *sgd* solver is separated. This node contains 135 poor and 215 good records and it is further split by the initial learning rate. As result, if the *sgd* solver was used with higher than 0.001 initial learning rate then 215 out of 280 settings were considered good. On the second level, the decision is also made by the solvers. If *tanh* or *relu* activation functions are used with the *adam* solver then about half of the settings will be considered good. On the other hand, if we use the *lbfgs* solver then no good settings were found.

Topology was included in the decision tree when its depth was increased. The *average neuron count in the layers* synthetic feature has got a lower limit. In other words, the more dense layers were used, more accurate the network will be. Therefore the usage of deep neural network in the future is suggested.

C. Best Parameters

Parameters of the top quantile has been analyzed in order to get some insight into the parameter choice. Records were considered as a set of training parameters. Frequent item set mining was used to determine which parameter pairs occur together. Apriori algorithm [1] was used to extract the frequent parameter pairs. The algorithm is based on a bottom-up approach and it requires a minimum support parameter. An item set is considered frequent if it contains at least as many elements as the minimum support value. Support is calculated for item sets as the proportion of the number of transaction containing each element of the set to the number of all transactions. In other words, the item sets with the higher support occur more frequently. Therefore the parameter pairs can be determined which usually train accurate neural network.

TABLE III
FREQUENT BEST PARAMETERS

Support	Item Sets
0.545685	(sgd)
0.538071	(relu)
0.449239	(tanh)
0.444162	(adam)
0.408629	(0.003)
0.403553	(0.001)
0.378173	(0.01)
0.327411	(0.03)
0.291878	(0.0001)
0.291878	(relu, sgd)
0.253807	(sgd, tanh)
0.246193	(adam, relu)
0.230964	(sgd, 0.01)
0.228426	(sgd, 0.003)
0.218274	(relu, 0.003)
0.218274	(relu, 0.01)
0.215736	((32, 64, 128, 64, 32))
0.213198	(sgd, 0.03)
0.210660	(relu, 0.001)
0.208122	(sgd, 0.001)

Table III shows the results of the frequent item set mining where minimum support was set to 0.2. There are 21 frequent item sets in the Table organized by their support value. The 10 most support item sets contain only a single element. The `sgd` solver and the `relu` activation function had the highest support. Next the `tanh` activation function and the `adam` solver achieved approximately 0.44 support. Then, the different learning rates (0.0001, 0.001, 0.003, 0.01, 0.03) form a group in the next six places. Finally, there is a single element item set in the 17th place which represents a topology.

The frequent item pairs are also found in Table III. The `sgd` solver with `relu` or `tanh` activation functions were the two most frequent setting pairs. The rest of the frequent pairs achieved less than 0.25 support and combine a learning rate with a solver or activation function. The most frequent settings trio was the `sgd, relu` with 0.01 learning rate and it had only 0.12 support.

Frequent item set mining of the good parameter settings confirmed the results of the decision tree. The `sgd` and `adam` solvers and the `relu` and `tanh` activation functions were

considered frequent. In addition, the apriori algorithm showed that the `sgd` solver is supported with both activation functions while the `adam` solver was considered frequent with only the `relu` activation function. Considering the initial learning rates, the 0.003 and 0.001 were the two most frequent.

D. Worst Parameters

Determining the frequently poor settings is also important in order to avoid unnecessary training. A parameter settings were considered worst if its score was lower than the 0.25 quantile of the scores. In this case, the high support advises against the usage of the settings. Also apriori algorithm was used to determine the frequent item sets which can be seen in Table IV.

TABLE IV
FREQUENT WORST PARAMETERS

Support	Item Sets
0.723350	(logistic)
0.505076	(lbfgs)
0.375635	(0.03)
0.365482	(0.0001)
0.362944	(0.01)
0.352792	(0.001)
0.337563	(0.003)
0.319797	((200, 400, 800, 400, 200))
0.271574	(logistic, 0.0001)
0.263959	(logistic, 0.01)
0.263959	(logistic, 0.03)
0.261421	(sgd)
0.256345	(sgd, logistic)
0.253807	(logistic, lbfgs)
0.251269	(0.003, logistic)
0.248731	(logistic, 0.001)
0.238579	((128, 256, 512, 256, 128))
0.233503	(adam)
0.213198	(logistic, adam)

Mining of poor parameter settings found 19 frequent item sets. The `logistic` activation function and the `lbfgs` solver got the highest support. Then different learning rate settings (0.0001, 0.001, 0.003, 0.01, 0.03) had similar support so their effect on the poor performance cannot be verified. Two relatively big network topologies were also found frequent. In addition, the frequent item pairs always contained the `logistic` activation function.

IV. DISCUSSION

Experimental results demonstrated the applicability of artificial neural networks for modeling the WiFi RSSI in indoor environment. The results were analyzed based on the R^2 coefficient of the regressor. Then, the training parameters were analyzed with statistical and data mining techniques in order to determine which parameters are beneficial for the training. Analysis showed that activation function and solver have the most significant effect on the performance of training.

Distribution of the achieved score showed three categories of training parameters. The first kind of settings performed around 0.0 which is considered poor. The score of the best settings was around 0.7 and about third of the records belonged

to this category. In other cases the score was between these two groups. This phenomenon could be caused by specific values of a given parameters. Therefore, these significant parameters and their beneficial settings should be identified in order to avoid unnecessary training.

Logistic activation function was always outperformed by the ReLU and tangent hyperbolic activation functions. Decision tree selected the activation function so this feature separates the training sets best. The selection criteria suggested that logistic activation function cannot achieve high accuracy for the given task. This suggestion was supported by the frequent item set mining because the ReLU and tangent hyperbolic functions were considered frequent in the top quantile of the trained parameters. In addition, the logistic function was frequent in the worst quantile. Consequently, the experimental results suggest the usage of ReLU and tangent hyperbolic activation functions.

Solvers also have a significant effect on the performance. Solvers was placed on the second and third level of the decision tree therefore it is also an important parameter. Experiments showed that the `sgd` and `adam` solvers could achieve the highest score in the given task. In the best results, these solvers were frequently paired with different initial learning rates. Although these solvers occurred between the parameters of the worst cases, they were less frequent. Hence, the experiments advise the usage of `sgd` and `adam` solvers.

Topology and learning rate were less significant for the training in the current experiments. Learning rate was considered frequent for the best cases albeit also frequent for the worst ones. The effect of learning rate could depend on solver and activation function. Topology was analyzed through its synthetic features which denoted the size of the layer and number of neurons. Increasing the depth of the decision tree showed that separation are made by the average minimum and maximum neuron counts in layers. Further experiments are needed in order to provide a detailed advice about the selection of learning rate and topology parameters.

In conclusion, the activation function and the solver are the two most significant parameter of the training for modeling WiFi RSSI in indoor environment. The `relu` or `tanh` activation function should be used with the `sgd` or `adam` solvers. The learning rate and topology have less effect on the performance. Experiments suggested some limitation to the size of the network. In the future, a wide range of networks should be evaluated with the fix solver and activation function.

ACKNOWLEDGMENTS

This research was supported by the grant EFOP-3.6.1-16-2016-00001 (Complex improvement of research capacities and services at Eszterhazy Karoly University).

REFERENCES

- [1] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
- [2] Abdul Halim Ali, Mohd Raziff Abd Razak, Muzaiyanah Hidayab, Syuwari Ashraf Azman, Mohd Zaim Mohd Jasmin, and Mohd Azmir Zainol. Investigation of indoor wifi radio signal propagation. In *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, pages 117–119. IEEE, 2010.
- [3] Paramvir Bahl, Venkata N Padmanabhan, Victor Bahl, and Venkat Padmanabhan. Radar: An in-building rf-based user location and tracking system. 2000.
- [4] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. Classification and regression trees. *wadsworth int. Group*, 37(15):237–251, 1984.
- [5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction, springer series in statistics, 2009.
- [6] Muzaiyanah Hidayab, Abdul Halim Ali, and Khairul Bariah Abas Azmi. Wifi signal propagation at 2.4 ghz. In *2009 Asia Pacific Microwave Conference*, pages 528–531. IEEE, 2009.
- [7] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2019-06-10].
- [8] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.
- [9] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.
- [10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [11] Hemant Kumar Rath, Sumanth Timmadasari, Bighnaraj Panigrahi, and Anantha Simha. Realistic indoor path loss modeling for regular wifi operations in india. In *2017 Twenty-third National Conference on Communications (NCC)*, pages 1–6. IEEE, 2017.
- [12] Donald F Specht. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576, 1991.
- [13] Zsolt Tóth and Judit Tamás. Miskolc iis hybrid ips: Dataset for hybrid indoor positioning. In *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 408–412. IEEE, 2016.

The Role of the Version Control Information in Code Comprehension

Tibor Brunner

Department of Programming Languages and Compilers
Eötvös Loránd University
 Budapest, Hungary
 bruntib@caesar.elte.hu

Zoltán Porkoláb

Department of Programming Languages and Compilers
Eötvös Loránd University
 Budapest, Hungary
 gsd@caesar.elte.hu

Abstract—Most software comprehension frameworks use the source code as the main resource for information retrieval. Advanced code comprehension process, however, requires the utilization of the full knowledge portfolio of the software system. In this paper we investigate how the version control information of the project can be utilized for extending our apprehension of large legacy systems providing a better understanding of the software under examination. We show that some of the hidden structural connections between the elements of the program can be revealed most easily by the development history of the system. A prototype implementation of the method using git version control information has been implemented as an open source extension of the CodeCompass software comprehension framework.

Index Terms—code comprehension, version control, git, software technology

I. INTRODUCTION

It is a well known fact, that the largest cost factor of the software products for their whole lifetime is the maintenance cost. One of the reasons is that prior to any maintenance activity – new feature development, bug fixing, etc. – programmers first have to locate the place where the change applies, have to understand the actual code, and have to explore the connections to other parts of the software to decide *how* to interact in order to avoid regression. All these activities require an adequate understanding of the code in question and its certain environment – something is currently impossible to automate, therefore the developers should pay their expensive time to carry out these actions.

Therefore, it is not a surprise that code comprehension is a key factor of modern software development, exhaustively researched by both the industry and academy. Various scientific and industrial papers published on the topic in conferences, e.g. in the series of International Conference of Program Comprehension, and in the Intellectual outputs No. O1 and O2 of the Erasmus+ Key Action 2 (Strategic partnership for higher education) project No.2017-1-SK01-KA203-035402: “Focusing Education on Composability, Comprehensibility and Correctness of Working Software” [1], [2] among others.

Most of the comprehension approaches are based on the source code. That is a logical approach, as the actual software might already diverged from the original specification and the documentation might also be over dated. Therefore, typical

comprehension tools analyse the source code, support fast navigation, feature location and reveal the internal structure of the software. However, not all the internal connections within the system can be detected by analysing the source. Virtual function calls on polymorphic objects, pointers, references, closures are among the program constructs where static analysis has limitations.

Code comprehension may not be restricted to existing code bases. Important architectural information can be gained from the build system, like relations between libraries, binaries and source files [3], [4]. Even more interesting structural connections can be revealed from the history of the project development: e.g. which files were added or changed at the same time, how these changes related to certain commit messages and which lines added/removed/changed frequently simultaneously.

CodeCompass is an open source code comprehension framework [5] developed by Ericsson and Eötvös Loránd University, Budapest, to help the code comprehension process of large legacy systems. The tool based on the LLVM/Clang compiler infrastructure [6], [7], and has been designed to be extremely scalable, seamlessly working with many million lines of code. Fast search options help locating the requested feature by text search. Once the feature has been located, precise information on language elements for variables, inheritance and aggregation relationships of types, and call points of functions are provided by the LLVM/Clang infrastructure. Easy navigation possibilities and a wide range of visualizations extend far more than the usual class and function call diagrams help the user the more complete comprehension. To make the comprehension more extensive, CodeCompass utilises the full portfolio of information available including build commands but also utilizes version control information, if available; git commit and branching history, blame view are also visualized.

In this paper we investigate the role of the version control information for code comprehension purposes. In Section II we overview the main categories of the existing comprehension software using a specific tool as an example. Section III describes the architecture of CodeCompass. In Section IV we show CodeCompass support for the version control information to reveal hidden connections between otherwise unrelated code segments. Our paper concludes in Section V.

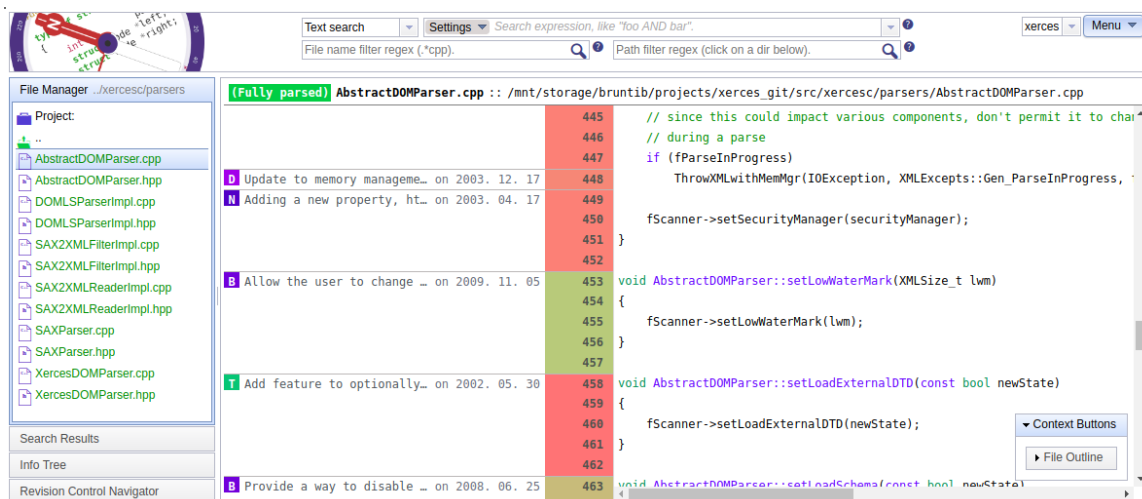


Fig. 1: Git change view

II. RELATED WORK

Code comprehension became a hot research topic recently, with dedicated user communities, proprietary and open source tools. On the software market there are several tools which aim some kind of source code comprehension. Some of them uses static analysis, others examine also the dynamic behavior of the parsed program. These tools can be divided into different archetypes based on their architectures and their main principles. On the one hand are tools having server-client architecture. Generally these tools parse the project and store all necessary information in a database. The (usually web-based) clients are served from the database. These tools can be integrated into the workflow as nightly CI runs. This way the developers can always browse and analyze the whole, large, legacy codebase. Also there are client-heavy applications where smaller part of the code base is parsed. This is the use case for IDE editors where the frequent modification of the source requires quick update of the database about analyzed results. In this section we present some tools used in industrial environment from each categories.

Woboq Code Browser [8] is a web-based code browser for C and C++. This tool has extensive features which aim for fast browsing of a software project. The user can quickly find the files and named entities by a search field which provides code completion for easy usability. The navigation in the code base is enabled through a web page consisting of static HTML files. These files are generated during a parsing process. The advantage of this approach is that the web client will be fast since no “on the fly” computation is needed on the server side while browsing. Hovering the mouse on a specific function, class, variable, macro, etc. can show the properties of that element. For example, in case of functions one can see its signature, place of its definition and place of usages. For classes one can check the size of its objects, the class layout and offset of its members and the inheritance diagram. For variables one can inspect their type and locations where they

are written or read.

OpenGrok [9] is a fast source code search and cross reference engine. Opposed to Woboq, this tool doesn’t perform deep language analysis, therefore it is not able to provide semantic information about the particular entities. Instead, it uses *Ctags* [10] for parsing the source code only textually, and to determine the type of the specific elements. Simple syntactic analysis enables the distinguishing of function, variable or class names, etc. The search among these is highly optimized, and therefore very fast even on large code bases. The search can be accomplished via compound expressions, containing even wildcards, furthermore, results can be restricted to sub-directories. In addition to text search there is opportunity to find symbols or definitions separately. The lack of semantic analysis allows Ctags to support several (41) programming languages. Also an advantage of this approach is that it is possible to incrementally update the index database. OpenGrok also gives opportunity to gather information from version control systems like Mercurial, SVN, Git, etc. OpenGrok has the ability to search not only in the content of source files but in their history as well. Since most of these version control systems (VCS) provide search functionalities in the project history (including commit messages and source files), OpenGrok can forward these queries to the given VCS. However, there are no extra visualizations in order to display the “blame view” so the developer could understand what other relevant changes happened in other files in the same commit. The branches of the history are invisible too. CodeCompass intends to support these use-cases.

Understand [11] is not only a code browsing tool, but a also a complete IDE. Its great advantage is that the source code can be edited and the changes of the analysis can be seen immediately. Besides code browsing functions already mentioned for previous tools, Understand provides a lot of metrics and reports. Some of these are the lines of code (total/average/maximum globally or per class), number of coupled/base/derived classes, lack of cohesion [12], McCabe

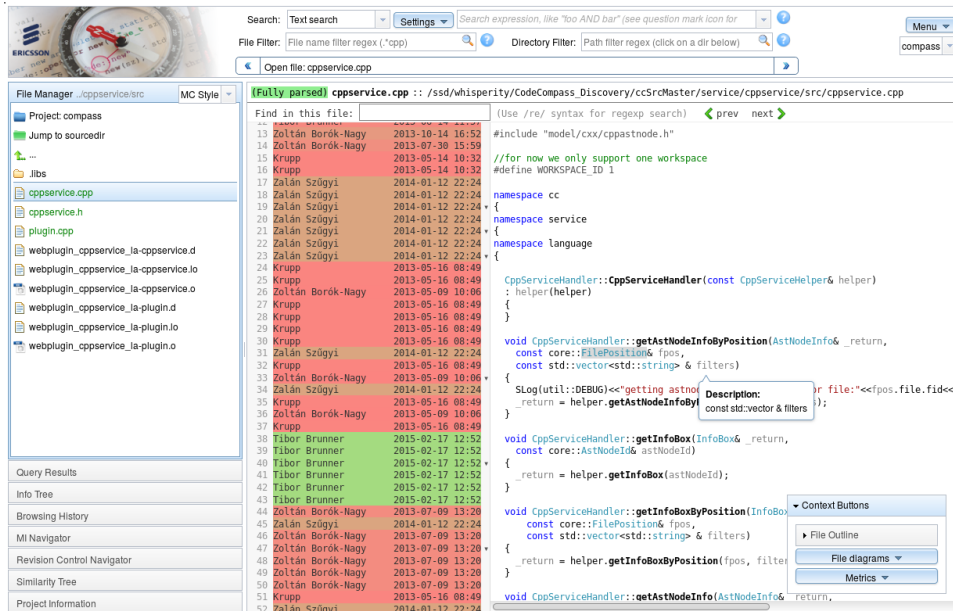


Fig. 2: Git blame view

complexity [13] and many others. *Treemap* is a common representation method for all metrics. It is a nested rectangular view where nesting represents the hierarchy of elements, and the color and size dimensions represent the metric chosen by the user. For large code bases, the inspection of the architecture is necessary. Visual representation is one of the most helpful way of displaying such structures. Understand can show dependency diagrams based on various relations such as function call hierarchy, class inheritance, file dependency, file inclusion/import. The users can also create their custom diagram type via the API provided by the tool.

CodeSurfer [14] is similar to Understand in the sense that it is also a thick client, static analysis application. Its target is understanding C/C++ or x86 machine code projects. CodeSurfer accomplishes deep language analysis which provides detailed information about the software behavior. For example, it implements pointer analysis to check which pointers may point to a given variable, lists the statements which depend on a selected statement by impact analysis, and uses dataflow analysis to pinpoint where a variable was assigned its value, etc.

Development tools The aforementioned tools are mainly designed for code comprehension. Another application area of static analysis is writing the code itself. This is a very different way of working in many aspects, which requires a slightly different tool set. Maybe the most widespread IDEs are *NetBeans* [15] and *Eclipse* [16] primarily for Java projects, and *QtCreator* [17] mainly for C++ projects. The recent open source tools tend to be pluginable so their functions can easily be extended according to special needs and domain specific tasks. The greatest benefit of these tools is the ability of incremental parsing, which means the real-time re-analysis of small deviations in the source code. The *Visual Studio* [18]

IDE has a rich interface for code comprehension features, like go to definitions and all references among others.

III. THE CODECOMPASS ARCHITECTURE

CodeCompass provides a read-only, searchable and navigable snapshot of the source code, rendered in both textual and graphical formats. CodeCompass is built with a traditional server-client architecture as depicted in Figure 3. The server application provides a Thrift [19] interface to clients over HTTP transport. The primary client that comes pre-packaged with the tool is a web browser based single-page HTML application written in HTML and JavaScript.

Since the interface is specified in the Thrift interface definition language, additional client applications (such as a command line client or an IDE plugin) can be easily written in more than 15 other languages supported by Thrift (including C/C++, Java, Python etc.). An experimental Eclipse plugin is already implemented.

A parsed snapshot of the source code is called a *workspace*. A workspace is physically stored as a relational database instance and additional files created during the *parsing process*. The parsing process consists of running different *parser plugins* on the source code. The most important parser plugins are:

A **search parser** iterates recursively over all files in the source folder and uses Apache Lucene [20] to collect all words from the source code. These words are stored in a search index, with their exact location (file and position).

The **C/C++ parser** iterates over a JSON compilation database containing build actions, using the LLVM/Clang parser [7] and stores the position and type information of specific AST nodes in the database. This database will be used by the *C/C++ language service* to answer Thrift calls regarding C/C++ source code.

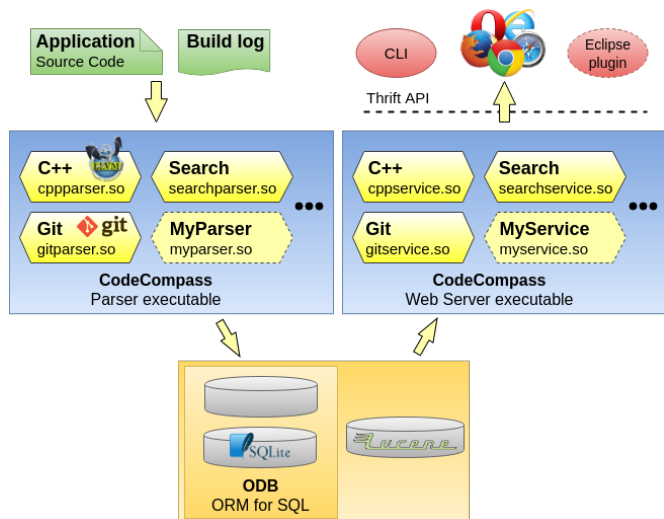


Fig. 3: CodeCompass architecture

The **Java parser** iterates over the same JSON compilation database containing build actions, using the Eclipse JDT parser [21] and stores the position and type information of specific AST nodes in the database. This database will be used by the *Java language service* to answer Thrift calls regarding Java source code.

Among other parser plugins the **Git parser** reads the version control information from the source tree (in the `.git` directories) and stores it into the project database.

CodeCompass has an extensible architecture, so new parser plugins can be written easily in C/C++ language. Parser plugins can be added to the system as shared objects.

On the **webserver**, Thrift calls are served by so-called *service plugins*. A service plugin implements one or more Thrift services and serves client requests based on information stored in a workspace. A Thrift service is a (remotely callable) collection of method and type definitions. All Thrift services have one implementation with the exception of the *language service*, which is implemented for C/C++, Java and Python. The *language service* is distinct in the sense that it provides the basic code navigation functionality for the languages it is implemented for. To put it simply, if this interface is implemented for a language, the user will be able to click and query information about symbols in the source code view of a file written in the given language.

Language service: This service provides query methods for symbols, for source files and globally for the workspace. It can return the symbol for a given source position, can return the corresponding node type and the corresponding source code fragment and documentation, the references to the symbol, diagrams – in GraphViz Dot format – or all information about the node in a tree format. For files, the service can return file level diagrams (in GraphViz Dot format), references to the file from other files, extended information about the file in a tree format, syntax highlighting data. This extended information includes, among others, all type and function definition within

the file, list of files this file is referred to and referred from. The service can also return all defined symbols in a workspace (such as namespaces, types, functions), so symbol catalogs can be implemented.

Search Service: This service provides 4 different type of queries: search in *text* for words, search among *symbol definitions* only, search among *file names* and suggest search phrases, based on a search *phrase fragment*. Text, definition and file search can be filtered by file name and containing directory. Definition search can be filtered for a certain language (C/C++, Java, JavaScript, Python, etc.) and symbol type (function definition, type definition, variable declaration, etc.) Developers can add additional service plugins delivered as shared objects in run-time.

The **Web-based user interface** is organized into a static *top area*, extensible *accordion modules* on the left and also extensible *center modules* on middle-right – see Figure 4.

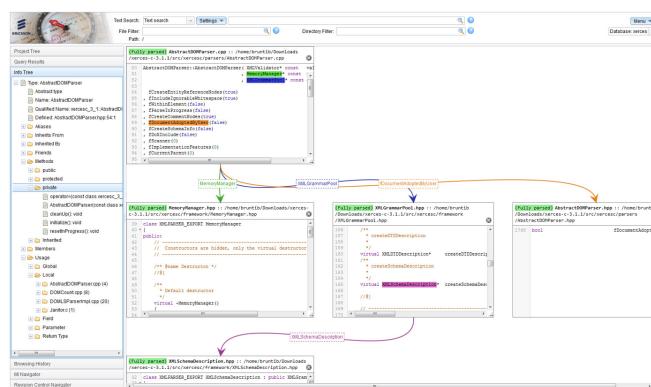


Fig. 4: CodeCompass user interface

The source code and different visualizations are shown in the center, while navigation trees and lists, such as file tree, search results, list of static analysis (CodeChecker) bugs, browsing history, code metrics and version control navigation is shown on the left. New center modules and accordion panels can be added by developers.

The top area shows the search toolbar, the currently opened file, the workspace selector, simple navigation history (breadcrumbs) and a generic menu for user guides.

IV. VERSION CONTROL SUPPORT IN CODECOMPASS

CodeCompass supports the code comprehension via various visualizations to present the different views of git information related to the project. As a starting point, one can initiate the **blame view** on any source code. Git blame view shows line-by-line the last changes (commits) to a given source file as seen on Figure 2. The background color of the committer also holds information: the commit that happened recently are colored lighter green, while older changes are darker red. This view is excellent to review why certain lines were added to a source file.

Clicking to the committers name of the blame view CodeCompass brings us to the **commit information**. This contains

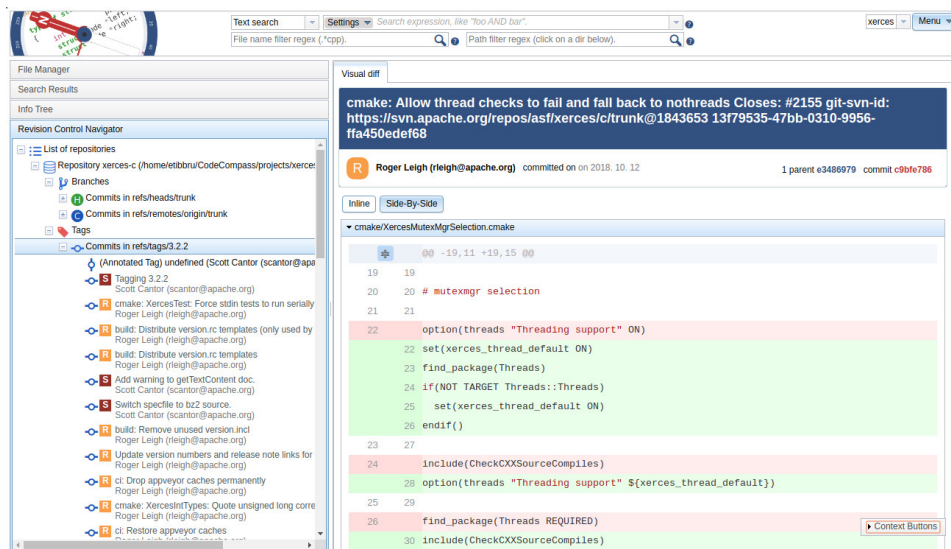


Fig. 5: Git branch view

the exact date and the message of the comment. Here we still can inspect the committed code. We see this on Figure 1.

CodeCompass can also show Git commits in a filterable list ordered by the time of commit. This search facility can be used to list changes made by a person or to filter commits by relevant words in the commit message.

Many times when we are reading the source code and find a fragment which is interesting in some ways or seems suspicious, we would also like to find what other parts of the current file have been modified at the same time. These colors are thus visual aid for determining which modifications belong to the introduction of the same new feature or bugfix.

Usually it is a project level decision whether the explanatory comments about the reasons of a modification should be incorporated in the source code or should be described in the commit message. The advantage of writing these comments in the source code is that this way these comments will be version controlled and also inseparable from the code. However, some information belong to the commit message of the modification (e.g. the issue which this commit solves, some links to external pages or earlier commits, etc.). In CodeCompass we would like to present these information to the user too, immediately at the currently displayed file and line.

In case of a software project the evolution of the program may also carry useful information. The organization of the development process is also up to the project members. One of the most common structures is to maintain a *master* branch which always contains the latest version of the project. When a new feature is created or a bugfix is being introduced then these are developed on a separate, so called *feature branch*. This branch contains one or more commits which make up the change together. When the introduced feature is stable on the feature branch then the changes are merged to the master. The bigger a new feature is, the more suitable it is to separate it on consecutive commits.

CodeCompass implements **branch view** that presents which commits have been developed on a separate branch and thus belong together. On Figure 5 we see a typical branch view.

All of the visualizations are based on similar graphical appearance of existing revision control tools, to minimize the cognitive effort for the developers when using revision control related information in CodeCompass.

Such use of revision control information of large legacy projects can reveal hidden connections in large software systems and help the complete comprehension of these projects.

V. CONCLUSION

Code comprehension is an important research area to support better understanding of large industrial software systems to reduce otherwise high maintenance costs. As most of the maintenance activities, like bug fixing, feature extension or modification require the full understanding of the system, comprehension is an expensive process requiring expert knowledge both from the domain and the implementation side. Tools supporting comprehension are mainly based on the static analysis of the source code. The structure of the system: class relationships, function call chains, and similar features can be determined with proper accuracy in this way.

However, software systems may contain hidden relationships between their components. Connections between configuration files and their application in the code, dynamic library usage and similar are hardly can be detected by static analysis. In the same time, those connections are likely reflected by the software development process, which can be retrieved from the version control information.

CodeCompass is an open source code comprehension framework which is intended to collect the whole information portfolio of the system under investigation. This includes not only the internal structure revealed from the source code, but also additional information, like the build and the git version control information.

The blame view shows the last committers line by line for the source lines visually expressing the “age” of the code. From here, the developer easily can navigate to the commit information, to check the commit message and the other files affected by the commit. One can also compare the code of different comments. Finally, we can inspect the development of the project by the traditional branch view. The applied visualizations inherit the graphical interface of the usual version control tools to make them familiar for the developers.

All these possibilities make the comprehension more complete thus in the further development process help to increase the code quality. The full implementation is available as open source at [22].

ACKNOWLEDGMENT

This work is supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

REFERENCES

- [1] T. Brunner, “Codecompass: an extensible code comprehension framework,” Eötvös Loránd University, Faculty of Informatics, Budapest, Tech. Rep. IK-TR1, May 2018.
- [2] C. Szabó. (2018) Programme of the winter school of project no.2017-1-sk01-ka203-035402: “focusing education on composability, comprehensibility and correctness of working software”. TUKE Kosice. Accessed 02-July-2019. [Online]. Available: https://kpi.fe.i.tuke.sk/sites/www2.kpi.fe.i.tuke.sk/files/personal/programme_of_the_first_intensive_programme_for_higher_education_learners_in_the_frame_of_the_project.pdf
- [3] R. Szalay, Z. Porkoláb, and D. Krupp, “Towards better symbol resolution for C/C++ programs: A cluster-based solution,” in *IEEE 17th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 2017, pp. 101–110.
- [4] —, “Measuring mangled name ambiguity in large c/c++ projects,” in *Zoran Budimac (Ed), Proceedings of the Sixth Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications. Belgrade, Serbia 2017.*, 2017. [Online]. Available: <http://ceur-ws.org/Vol-1938/paper-sza.pdf>
- [5] Z. Porkoláb, T. Brunner, D. Krupp, and M. Csordás, “Codecompass: An open software comprehension framework for industrial usage,” in *Proceedings of the 26th Conference on Program Comprehension*, ser. ICPC ’18. New York, NY, USA: ACM, 2018, pp. 361–369. [Online]. Available: <http://doi.acm.org/10.1145/3196321.3197546>
- [6] The LLVM Compiler Infrastructure. <https://llvm.org/>. [Online]. Available: <https://llvm.org/>
- [7] Clang: a C language family frontend for LLVM. <https://clang.llvm.org/>. [Online]. Available: <https://clang.llvm.org/>
- [8] Code Browser by Woboq for C and C++. <https://woboq.com/codebrowser.html>. [Online]. Available: <https://woboq.com/codebrowser.html>
- [9] OpenGrok. <https://opengrok.github.io/OpenGrok>. [Online]. Available: <https://opengrok.github.io/OpenGrok>
- [10] Exuberant CTAGS. <http://ctags.sourceforge.net>. [Online]. Available: <http://ctags.sourceforge.net>
- [11] SciTools: Understand. <https://scitools.com>. [Online]. Available: <https://scitools.com>
- [12] B. Henderson-Sellers, *Object-oriented Metrics: Measures of Complexity*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [13] McCabe, “A complexity measure,” *IEEE Transactions on Software Engineering*, vol. 2, pp. 308–320, 1976.
- [14] CodeSurfer. <https://www.grammatech.com/products/codesurfer>. [Online]. Available: <https://www.grammatech.com/products/codesurfer>
- [15] Apache NetBeans. <https://netbeans.org>. [Online]. Available: <https://netbeans.org>
- [16] Eclipse. <https://www.eclipse.org/ide/>. [Online]. Available: <https://www.eclipse.org/ide/>
- [17] Qt Creator. <https://www.qt.io/>. [Online]. Available: <https://www.qt.io/>
- [18] Visual Studio. <https://visualstudio.microsoft.com>. [Online]. Available: <https://visualstudio.microsoft.com>
- [19] Apache Thrift. <https://Thrift.apache.org>. [Online]. Available: <https://Thrift.apache.org>
- [20] Apache Lucene. <https://lucene.apache.org/core/>. [Online]. Available: <https://lucene.apache.org/core/>
- [21] Eclipse Java development tools (JDT). <http://www.eclipse.org/jdt>. [Online]. Available: <http://www.eclipse.org/jdt>
- [22] CodeCompass website. <https://github.com/Ericsson/CodeCompass>. [Online]. Available: <https://github.com/Ericsson/CodeCompass>

The URLs were retrieved on 8 September 2019 if other date is not given.

Teachers' attitudes towards implementing coding at schools

Hana Bučková

Department of Technical Education and
Information Technology
Faculty of Education
Palacký University
Olomouc, Czech republic
hana.buckova02@upol.cz

Jiří Dostál

Department of Technical Education
and Information Technology
Faculty of Education
Palacký University
Olomouc, Czech republic
j.dostal@upol.cz

Abstract — The process of implementing coding at schools attended by pupils from six to fourteen years of age is a current trend in developed countries. In some cases, simple activities involving coding might be even included at kindergartens attended by children from three years of age. The crucial aim is to develop the so-called computational thinking which is going to be more and more essential for a successful personal and professional development of every individual.

There is a broad range of factors influencing the success of the change in the curriculum, which is, *inter alia*, supposed to include coding. In addition to the necessary teaching aids and know-how, teachers also need to have positive attitudes towards its implementation. The main objective of our research is to identify teachers' attitudes towards introducing coding to their classes. We discovered that not all teachers of computing agree with including algorithmization and coding in their teaching. Approximately one half of the questioned educators hold a negative attitude towards the curricular change; they would rather continue teaching computer user skills and basic programs. Their attitude proves to be an impediment to a successful curricular change as teachers should fully grasp the teaching content and communicate it with enthusiasm to their pupils. Insufficient knowledge is the key factor influencing teachers' negative attitudes towards the introduction of coding to schools.

Keywords— coding, school, teacher, computational thinking, pupil

I. INTRODUCTION

Contemporary modern society, which is developing dynamically not only from the technological point of view but also from the cultural, social and economic level, emphasizes today's young generation to develop their knowledge in the field of information thinking due to their further application in both professional and private life. As a consequence of these changes, curriculum innovation, teachers' access to newly introduced topics and their implementation in teaching are necessary. At present, the content in the educational area of Information and Communication Technologies is innovated, based on revisions made by the Ministry of Education, Youth and Sports. The skills which pupils acquire in studying ICT at the user level need to be transformed to the level of development of digital literacy and creativity of pupils.

The curriculum change also carries new demands on teachers, teachers' access to the planned change, and their expertise. Teachers' approaches to curricular reform are crucial and their views on changing ICT education are important. On the research basis their opinions on new topics of the subject of Informatics are examined.

In the Czech Republic, the state of Computer Science education is not optimal. From the results of the Czech School Inspectorate, lessons are provided by untrained teachers. The opinions of qualified or unskilled teachers are crucial, as the teacher is a direct actor of change. They are supposed to decide what form the newly designed curriculum transforms into real education will be implemented as well as what the curriculum will look like and what qualities the achieved curriculum will represent.

II. THE ONGOING CURRICULAR REFORM OF COMPUTER SCIENCE EDUCATION IN THE CZECH REPUBLIC

At present, Information and Communication Technologies at primary schools in the Czech Republic are taught in a time allowance of one hour at primary schools and one hour at lower secondary schools [1]. Pupils learn user skills such as working with a text editor (Word), spreadsheet editor (Excel), creating a PowerPoint presentation, handling e-mail, and much more. This teaching is obsolete and is not focused on the current trend, i.e. teaching algorithmization and programming, i.e. to develop information thinking and digital literacy.

In the Czech Republic, changes are being prepared, after which education should be more integrated with the digital world, and thus respond to technological developments. The National Institute for Education is preparing new educational programs (see MEYS, 2017), where one of the objectives will be the development of information thinking. Due to the development of technologies, many of the initiatives have ceased to be up-to-date and for these reasons, the innovation of the ICT curriculum is necessary. A change in this area has been solved by the National Institute for Education since May 2016, when the concept of development of digital literacy and information thinking was prepared and new educational content was proposed.

On August 30, 2018, the Ministry of Education announced, in accordance with the Education Act on

pre-school, primary, secondary and tertiary professional education, an experimental verification entitled "Development of Computer Thinking of Children and Pupils in Kindergartens, Primary and Secondary Schools". Experimental testing began on September 1, 2018 at selected nursery, primary and secondary schools, where the selection of schools was entrusted to faculties of education. The selection of schools was approved by the Ministry of Education. The verification is conducted by the Ministry of Education, Youth and Sports in cooperation with the National Institute for Education and is supported by faculties of education involved in the project [3].

Verification is carried out in two phases:

1. The phase was materialized in the school year 2018 - 2019
2. The phase is materialized in the school year 2019 - 2020

The expected learning outcomes are:

- Programming and algorithmization
- Informatics (other topics)
- Introduction to robotics [4]

There are newly created textbooks for lower secondary schools:

- Scratch programming for primary school
- Programming in Scratch II (projects for lower secondary school)
- Introduction to Informatics for grammar schools
- Work with data for 5th to 7th grade of grammar schools
- Robotics at lower secondary school with LEGO® Mindstorms [4]

Educational materials are currently being tested at selected grammar schools. They will be further modified in this school year, and in 2020 their definitive version will be published, which should serve for teaching the given topics at grammar schools.

III. OPINIONS OF PILOT SCHOOL TEACHERS ON A NEW CONCEPT OF COMPUTER SCIENCE EDUCATION

The teacher is a key element in shaping the final form of the curriculum which is implemented in the teaching of computer science at grammar schools. However, a number of factors which deliberately and randomly enter into the process of curricular design at school level and they influence it and form it. In particular, it is necessary to identify with the Framework Educational Program (FEP). At the same time, it is desirable to take into account local conditions and internal factors standing on the teacher's side - especially the time possibilities of preparation for teaching and his or her didactic skills. The specific content of education is formed with regard to the expected competences of pupils, which are linked to the level of technological and social development. It reflects the problems and life situations that pupils need to be prepared for. They are also subject to continuous changes and the content of education must reflect them, otherwise education would be ineffective. Neumajer (2016) argues: „It is evident that updates are vital because of current fields of educational areas ICT in FEP has not changed since 2004.“ [5]

A. Opinions of pilot school teachers on innovation of Computer Science education

The teaching of computer science will undergo a radical change, where students will work with data and information, learn the basics of programming and the development of computer thinking. User skills will be taught in other subjects. In the subject of informatics, pupils learn logical thinking and problem solving [6].

In the school year 2018/2019, the first phase of verification took place at approximately 70 schools throughout the Czech Republic. Currently, the second phase of pilot testing is underway within the project Support for the Development of Computer Thinking, where the beta versions of educational materials corresponding to the new FEP of the subject Informatics are verified. In the article we present the opinions of teachers who were involved in the project, how they perceived the content of textbooks and how the pupils reacted to prepared textbooks. Teachers of all pilot schools provided ongoing feedback and, at the end, prepared so-called reflective reports, which were in many aspects close to each other. Below are representative views of 3 selected teachers.

Pilot school teachers' view

Do you think that the curriculum that you are new to computer science is useful for their future career?

Teacher No. 1: This is useful for the future profession because it forces pupils to think mathematically. Scratch uses their mathematical imagination, algorithmization. For some pupils, this is very difficult and sometimes impossible. I think some will not use it (future bricklayers and community service workers do not need to know how to program in Scratch).

Teacher No. 2: I am not sure whether the curriculum is useful in its content, but certainly in terms of developing logical thinking.

Teacher No. 3: Yes, whatever it may be. They teach to think, make decisions, look for other solutions, situations where the chosen solution may not work, boundary conditions.

Do you think that the curriculum that you and your pupils are newly discussing in computer science is useful for their everyday life?

Teacher No. 1: I would say that for some pupils it is unnecessary for their daily lives. They won't take full advantage of it. Of course, to determine some algorithm activity, the algorithm will solve the problem. But it doesn't have to be strictly processed in Scratch programming.

Teacher No. 2: As in the previous answer, I think it is appropriate for pupils to develop logical thinking. However, it is not appropriate to use the program as a supplement to their leisure time.

Teacher No. 3: Yes. It also supports creativity and skills to search for new solutions (not just using CTRL C, CTRL V keys or copying from the board). Moreover, they are not bored. They have a feeling that they can relax between lectures. And they also feel that they relax before „difficult lectures“.

What problems did you have to solve in relation to the realization of teaching according to the new textbooks? Select only the most important ones.

Teacher No. 1: This is a challenge for some pupils. A few pupils got really interested in and worked at home alone,

trying new attitudes. But I would take it as an exception (1-2 pupils per class). Then there are pupils who do not enjoy it. They keep asking me what they need this for. I have to motivate them. They most liked Chapters 6 and 7, where there were few methodological sheets and games. Some of them had difficulty working according to the pupil's papers, I had to explain what the authors wanted from them. Some 70 % of pupils understood the task, nevertheless, 30 % rather had a problem with their desire to work. They simply did not want to.

Teacher No 2: Demanding software (ie. support).

Teacher No. 3: Cancelling of lectures (I would mind this regardless the content of lectures). I had to skip the prepared plan of lectures.

Do you think that the curriculum contained in the tested textbook is suitable for inclusion in teaching at all schools throughout the Czech Republic?

Teacher No. 1: I am afraid that teachers who do not have a positive attitude to it will be perplexed and that they will discourage most students. The curriculum contained in the textbook could be taught at grammar schools, but in my opinion, this way:

a) There should just be max. 5 lectures of ICT per week, so that pupils could experiment in lectures.

b) Advanced skills should be taught at facultative subjects (Informatics at our school).

Teacher No. 2: Definitely not. I have experienced teaching of ICT at different types of schools (special schools, schools for challenged pupils, schools for a high number of socially challenges pupils etc.) and it is hard to imagine to teach this subject at some of them. In my eye, this subject is suitable just for gifted pupils.

Teacher No. 3: Yes. It is accessible for everyone who would like to try it. And it is for free.

Did you have troubles to motivate pupils? Were lectures monotonous?

Teacher No. 1: I tried to motivate my pupils. I always focused on what they might like. We discussed this in detail and created projects. Others topics, which the pupils did not find attractive, were not analyzed exhaustively. The worst chapter was Coordinates. The theme of parameters have not been not covered yet. I tried to include the creation of new projects (as described in the evaluation).

Teacher No. 2: Yes, we have still been dealing with this problem. Lectures are way too long. Pupils are rather reluctant to solve similar problems again and again.

Teacher No. 3: No. No.

What do you think of the amount of the curriculum? Is it optimal?

Teacher No. 1: I have a feeling that the amount of the curriculum is optimal. I would not teach the full volume of curriculum all pupils in the Czech Republic, but only those pupils who choose this subject as a facultative one. Withing this subject and would explain topics elaborately and I would take up programming with Micro:bit and Ozobot programming, as well as Turtle graphics etc.

Teacher No. 2: As I see it, the amount of curriculum was too large. As a result, we were not able to cover the topics thoroughly.

Teacher No. 3: We have not had enough time to cover all the topics, and I am sure we will not have enough time to cover all the topics in future either. We might barely cover Chapter No. 8.

Do you need a course to better understand how to teach the curriculum content?

Teacher No. 1: I did not face big troubles. I just needed to learn how not to tell the correct answer as well as how to lead pupils. At the very beginning the pupils seemed a little unhappy because they desired to know the correct results.

Teacher No. 2: Yes, I went to the training provided by Fraus Publishing and I must admit that it helped me very much in working with Scratch. "Getting through" the beginning of the program was much easier.

Teacher No. 3: No, the methodology is greatly made. I really appreciate suggestions to start up discussions in classes.

What do you think is the benefit of the new curriculum contained in the textbook?

Teacher No. 1: As I mentioned before, mathematical thinking and algorithmization.

Teacher No. 2: As I said, it is beneficial for development of logical thinking, creativity, fantasy, and imagination.

Teacher No. 3: It develops logical thinking and pupils have no resistance to this form. They are not afraid of it, unlike, for example, mathematics. In this case they may feel rather irritated or even frightened.

Do you think that the curriculum for the control of user programs can be transferred to the teaching of other subjects (e.g. Word to Czech Language, graphics to Art, Excel to Mathematics) and taught by teachers of these subjects? Will they be able to do that? Will there be enough time for this?

Teacher No. 1: Teachers will have not time for it. They will not be able to teach this way (there are senior teachers in our school and I help them use Word and Excel programs).

Pupils need to use computers. It is crucial for them to see Excel in the classroom on the datasheet. Furthermore, the amount of 27-30 pupils in the classroom is absolutely unnecessary.

I do not want to underestimate any teachers, but generally, it is not possible. Certainly there will be some teachers (for example of Czech Language, Maths or Art) who could master this brilliantly, but most of them leave it for the end of the year, and then say there was no time for it.

Word, Excel a Power Point should be part of ICT classes. Mastering these programs is important for companies. In fact, nearly everybody needs to know how to use these programs recently. Even a builder is in need to write a letter.

The field of computer graphics does not seem to be of such an importance. This might be part of facultative subjects.

Teacher No. 2: Yes. I think it is definitely possible and that it is already happening in some schools. Teachers grow along with the times, and I believe that many of them use user programs to facilitate learning for students and themselves.

Teacher No. 3: No. No one has time or space, most of teachers do not even have any kind of knowledge. It seems like a requirement for Mathematics to dissolve into subjects like Physics, Chemistry, or Economics, so that each of the teachers of those subjects teaches what his or her science needs.

Based on the system which we have practised, I think that we should provide a sort of "service" and teach pupils the basics of working with data, formatting text, using spreadsheets. What is more, we ought to convince our colleagues to use this service.

Of course, I also want to learn some form of programming, I like this course pretty much as I like working with Lego robots.

I really do not know what to do in this current situation. I wish I had 2 lectures per week during all four years. I would cover particular parts, including user controlling, however, in a universal way. I oppose teaching based on simple explaining of elementary functions of Word and saying that a particular version of Word works this way and some keys like Ctrl Alt have different functions in another version of Word.

I would like to use my colleagues to practice the learned procedures, for example, teachers of Czech Language require homework essays written on PC with typographical rules, or seminar work in all school subjects, like Laboratory, meet the common requirements for larger work, using graphs and functions of Excel.

I have some other requirements, too. I also want children to write by hand, to touch pencils, brushes, paints, clay in art education. Secondly, I would like them to build a real electrical circuit, crane or something else, ie. to be able to create their own text and not to copy everything. Finally, I wish they were able to talk live, argue, discuss with facts, and not hide behind electronic communication.

Pilot teaching has proven to be appropriate for the age and the possibilities of the pupils, but it depends very much on the didactic concept of the curriculum, which is a tool for achieving the expected learning outcomes. This has been significantly helped by the developed textbooks, which have become a key source of information for teachers in pilot lessons. This made it possible to induce stimulating educational situations.

The authors' approach, which was used in the creation of teaching materials, can be described as responsible. It was based on their erudition and already during the creation of the teaching materials were partially tested in school practice (always at two schools). This allowed learning tasks to be better adapted to the pupil's level of thinking. Teaching materials can be described as successful. Nevertheless, strengths and weaknesses were revealed during the verification process. The empowerment of authors as a teacher for whom the curriculum is new and meets him or her for the first time can be described as the strongest. The tasks are well explained. However, there are cases where it will be desirable to allow teachers to complete training. These are currently being prepared within the project and are piloted in the 2019/2020 school year.

IV. TEACHERS' POINT OF VIEW ON THE FORM OF CURRICULUM, OR WHAT WOULD THEY LIKE TO TEACH?

In addition to pilot testing, empirically based research is underway, with the focus being on finding out what is happening on the teacher side in relation to curricular reform. As a rule, school reforms are triggered by tensions caused by an unsatisfactory state in educational reality, such as the existence of outdated curricula. This was also the case in the Czech Republic with regard to the subject of Informatics. Subsequently, expert discussions took place

and a new curriculum was proposed and is now being experimentally implemented at selected schools. Binding documents therefore prescribe the curriculum to be the subject of education in schools. But what is the idea of teachers themselves? What do they think should be taught in schools?

The aim of the research was to find out the teachers' opinions on the subject matter of informatics, so that we know whether the new conception of teaching will be in line with their ideas. Examined teachers were asked questions about current topics in computer science teaching, and new topics to be implemented as part of curriculum changes.

As part of the research, 320 primary schools in the Czech Republic were approached on the basis of random sampling. Out of them, 72 teachers of Informatics teaching at lower secondary schools responded positively regarding their participation in the research. Out of these 72 teachers, 33 were qualified in teaching of Computer science or a related field, while the remaining 39 were certified in other subjects, but still teaching Computer science. The topics of Computer science education related to the ongoing curriculum innovation of the FEP were presented to the examined teachers. The results of the research are interpreted by means of cluster analysis.

Cluster analysis is a set of methods that make it possible to search groupings of similar objects in empirical data [7]. At the same time, the characteristics of the clusters and their number are not known in advance - they must be derived from research data [8]. These groupings can be described, for example, by type profiles in a number of monitored variables, by objects which are merged into the grouping, and a number of other characteristics. The cluster analysis or clustering is usually performed on a set of objects, each of which is described by the same set of characters which it makes sense to follow on the set of objects [9]. Therefore, if there are several teachers with similar opinions in the sample, they form a cluster.

Euclidean distance were used to form clusters

$$D_{ij} = \sqrt{\sum_{k=1}^K}$$

and Ward's method (Ward's Minimum Variance Clustering Method) [10].

Reliability was verified by the so-called Cronbach coefficient alpha. It is a calculation of the internal consistency of the questionnaire, which determines its reliability. To confirm this calculation we used the Split-half method, where the results of the sorting are divided into even and odd and the correlation is determined between the two groups, cf. [11].

Split-half method

The total scores for all respondents are calculated for the two halves of the test and then the correlation between these total scores is calculated.

$$\rho = \frac{2|r|}{1 + |r|}$$

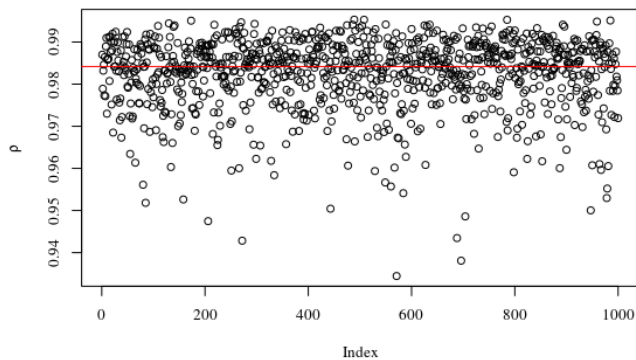


Fig. No. 1 Bootstrap ρ reliability calculation with median indicated.

The results of the reliability analysis confirmed the reliability of the questionnaire (bisection method: 0.9842 - median; 0.9824 - mean. In both cases the value is higher than the minimum required limit of 0.70.

A graphical representation of the average values of the investigated q-types broken down by identified clusters also seems to be useful.

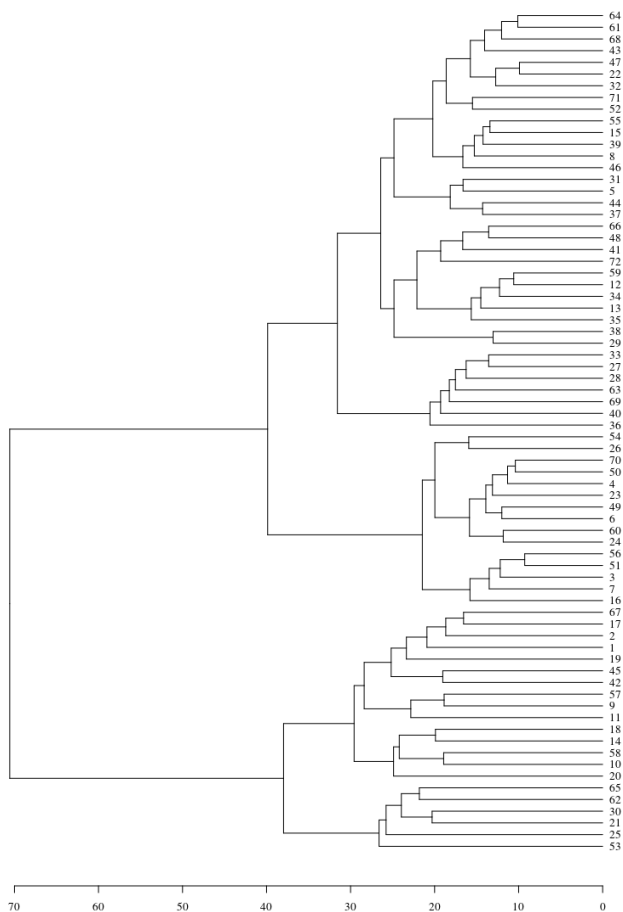


Fig. No. 2: Fig. Dendrogram: Clustering of teachers

By means of cluster analysis it was verified whether it is possible to divide the respondents into partial characteristic groups. We found three groups of respondents - teachers (3 clusters). Each of these groups evaluates the submitted Q-types differently.

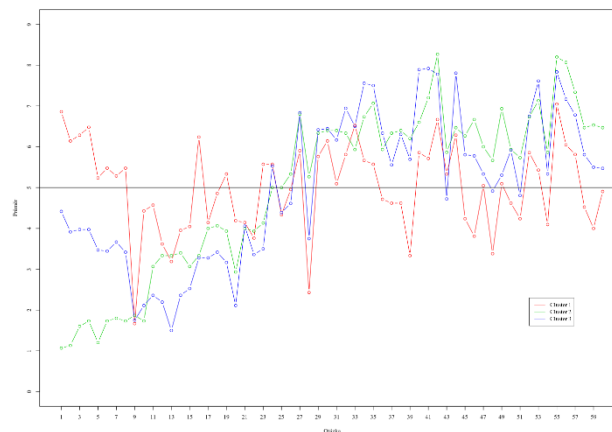


Fig. No. 3: Graph of average response in clusters

Cluster No. 1 – “Supporters of a thematically balanced development of computer skills and digital literacy”

Teachers included in the first cluster prefer balanced topics that can be included under the term "digital literacy", ie. working with text and spreadsheets, work safety, computer graphics, creating presentations, basic computer operations, etc. and topics related to programming and algorithmization. In other words, they are of the opinion that competences related to the use of information technology in everyday and working life are very important for pupils. In addition, however, they are fully aware that technology is increasingly affecting our lives, to the extent that there are gradual changes not only in industry, but also in other areas of human action, and we are increasingly witnessing automation, robotization and artificial intelligence. That is why it is necessary to transform education at grammar schools.

However, it should be noted that even in this group the emphasis on programming and algorithmization is not too high, as the topics related to it do not appear in the most highly rated positions. They are evenly mixed with topics related to the development of digital literacy. That is why we refer to this group as “supporters of a thematically balanced development of computer skills and digital literacy.”

Cluster No. 2 – “Opposers of algorithmization”

Based on the research, we found that there is a group of teachers who prefer to teach topics related to the general use of computers in everyday life. They put great emphasis on teaching topics such as security, working with text and spreadsheets, creating presentations, and mastering computer skills. Furthermore, work with graphic editors, the Internet, mastering the principles of communication by e-mail or work with peripherals. In general, these topics can be included under the notion of “digital literacy development” or “user competence development”. However, this is also considered important by the teachers in Cluster No. 1.

However, what makes this cluster different is that it very strongly rejects the topics associated with algorithmization. Therefore, this group can be aptly described as “opposers of algorithmization”. It is also remarkable that it is the smallest, but not negligible, group of teachers.

Cluster No. 3 – “Opposers of programming”

The last, but numerically most extensive cluster of teachers, is very similar to Cluster No. 2 in terms of topic preference. They prefer to teach topics related to the general

REFERENCES

use of computers in everyday life. However, they are teachers who strongly reject programming topics.

V. CONCLUSION

The Czech Republic seeks to strengthen the teaching of Informatics, or as the case may be, a curricular reform. However, we know from experience abroad as well as previous knowledge of introducing innovation into schools that there may be a number of obstacles to positive change.

The new concept of informatics teaching must be in line with the reality of the schools where the curriculum will be implemented, not only in terms of the conditions for teachers themselves, but also of the appropriate technical equipment in schools. The prescribed curriculum should coincide with the views of teachers who are designers of SEPs. [12] and curriculum implementers. A situation in which teaching is designed and implemented by a teacher who does not identify with the prescribed curriculum cannot be described as optimal. Teachers are not fully identified with the new concept and many of them are unable to deliver the newly designed educational content. On the one hand, it seems to be a problem that they do not have the necessary professional knowledge and they also have no idea how to manage the curriculum in a suitable way. It will be necessary to make proposals for measures aimed specifically at this group of teachers, in order to mitigate negative attitudes, and thus teachers would also be able to implement this teaching.

ACKNOWLEDGMENT

The paper was created with a financial support of the project of Grant fund of the Dean of the Faculty of Education, Palacký University Olomouc, 2019 in the framework of the project "Curricular Innovations of the Subject of Computing in the Context of the Teachers' Opinion and Research of Optimal Preparation of Future EFL Teachers for the Use of Information Technologies in Class".

- [1] MŠMT. (2005). Rámcový vzdělávací program pro základní vzdělávání. 126 s. Dostupné z: <http://www.nuv.cz/t/prehled-uprav-rvp-zv-1>
- [2] MŠMT. (2017). Rámcový vzdělávací program pro základní vzdělávání. 164 s. Dostupné z: <http://www.nuv.cz/t/prehled-uprav-rvp-zv-1>
- [3] MŠMT. (2018). Pokusné ověřování Rozvoj informatického myšlení dětí a žáků v mateřských, základních a středních školách. Dostupné z: <http://www.msmt.cz/vzdelavani/zakladni-vzdelavani/vyhlaseni-pokusneho-overovani-rozvoj-informatickeho-mysleni?highlightWords=Vyh%C3%A1%C5%A1en%C3%AD+pokusn%C3%A9ho+ov%C4%9B%C5%99ov%C3%A1n%C3%AD+rozvoj+informatick%C3%A9ho+my%C5%A1len%C3%AD+mate%C5%99sk%C3%BDch+z%C3%A1kladn%C3%ADch+st%C5%99edn%C3%ADch+%C5%A1kol%C3%A1ch>
- [4] Informatické myšlení. (2019). Učebnice a vzdělávací materiály. Dostupné z: <https://imysleni.cz/ucebnice>
- [5] Neumajer, O. (2016). Jak se bude zavádět informatické myšlení a zvyšovat digitální gramotnost ve školách. Praha: Wolters Kluwer ČR a. s., roč. 13, č. 11, s. 20-2.
- [6] České školství čeká revoluce v informatice. Děti se budou učit programovat a logicky myslet. (2018). Dostupné z: <https://connect.zive.cz/clanky/ceske-skolstvi-ceka-revoluce-v-informatice-deti-se-budou-ucit-programovat-a-logicky-myslet/sc-320-a-195437/default.aspx>
- [7] Osecká, L. (2001) Typologie v psychologii : aplikace metod shlukové analýzy v psychologickém výzkumu. 1. vyd. Praha: Academia. 161 s. ISBN 80-200-0854-3.
- [8] Chráška, M. (2003). Úvod do výzkumu v pedagogice. 1. vyd. Olomouc: PdF UP, 198 s. ISBN 80-244-0765-5.
- [9] Lukášová, A. & Šarmanová, J. (1985). Metody shlukové analýzy. 1. vyd. Praha: SNTL, 212 s. ISBN neuvedeno.
- [10] Murtagh, F. & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of Classification*, 31, 274–295. doi: 10.1007/s00357-014-9161-z.
- [11] Bonett, Douglas & A. Wright, Thomas. (2014). Cronbach's alpha reliability: Interval estimation, hypothesis testing, and sample size planning. *Journal of Organizational Behavior*. 36. 10.1002/job.1960.
- [12] Pexa, P., & Čapková, M. (2018). Inovace RVP a ŠVP v závislosti na aktuálních trendech výuky ICT na ZŠ. *Journal of Technology and Information Education*, 10(2), 73-93. doi: 10.5507/jtie.2018.012.

New Challenges in Forensic Analysis in Railway Domain

Jasmin Cosic¹, Christian Schlehuber², Drazen Morog³
Information Security Department
DB Netze AG
Frankfurt am Main, Germany

¹jasmin.cosic@deutschebahn.com, ²christian.schlehuber@deutschebahn.com, ³drazen.morog@deutschebahn.com

Abstract— Traditionally, the railway system was closed, vendor-dependent, old-fashioned with an accent on safety. Digitalization process has not bypassed the railways domain and with new upcoming technology and new functionality in railway domain (digital interlocking system, automated train operation, object recognition, GPS satellites localization, artificial intelligence, predictive maintenance, etc.), traditional railway domain has got a vulnerability that can be exploited.

In this paper, authors will deal with this “new” upcoming railway technology and challenges in digital investigation process, in such kind of environment.

Keywords—digital forensic, railway security, chain of custody, digital evidence, cybersecurity in railway, digitalization

I. INTRODUCTION

The process of digitalization, which is ongoing in the world, has not bypassed the railways - a domain that until now has been traditionally closed and there were no dangers posed by new technologies. A simple digitalization trend involves adapting existing technologies to make them more accessible and accessible to users, and this is possible with the use of new ICT solutions. Traditionally, existing solutions have been implemented in the railroad by the “vendor-dependence” concept and are based on the principles of the black box, and are characterized by, conditionally speaking, complete closure to external systems. The architecture of a railway is usually made up of three levels, namely: the first level is the low-element area or field element area components (FEAs)- the train detection system (TDS), signals and lineside electronic unit (LEU). The second level consists of interlocking systems with a technology center, which are, from a safety point of view, most important because they control the trains - they determine which tracks the train will run on. Object controllers are components that are driving FEAs (interface between FEAs and interlocking). The third level is the OC - Operating Center from which the entire functionality of the railway is monitored, and which is traffic supervisor and manages interlocking systems.[1] With the process of digitization, a new era begins, and the railway is completely transformed, so that a traditional, "safety-oriented system" is slowly opened, modernized and becomes more "security-oriented". This process of "modernization" involves the use of CoTS (Commercial of the Shelf) solutions, the use of no more private but more "public", open networks, and standard protocols (TCP/IP, for example) with standard applications and hardware solutions (standard ICT). The goal was certainly, first, economical business, increasing the quality of

services provided, getting closer to customers, different services- P2P (people to people), M2M (machine to machine), P2M (people to machine) that enable interconnectivity, but also reduce the cost of introducing and maintaining new services. Of course, this also has its price, which can be shown as a risk of digitization.

II. DIGITALISATION PROCESS IN RAILWAY

A. Digital Transformation Levers

The notion of ETCS and ERTMS is closely linked to the digitalization process in the railways, while some EU countries have their own plans and specific digitization programs that must be compatible with ETCS for the simple reason of interoperability. The term ETCS stands for (European Train Control System) and ERTMS (European Rail Traffic Management System), for example, Germany has its own digitization plan called NeuPro (Neue Produktion) and DSTW (Digitale StellWerke). The central place and the most critical part of every railway system is the interlocking system, and there are several generations of them - the so-called mechanical (from the time of emperors), electro-mechanical, relay, electronic, and up to today with the modern digital interlock systems. The intention with the new plans for modernization and digitization is to move from all these generations that we have today to the new digital systems.[2] Along with the modernization of the interlocking system, goes modernization of communication, so the plan is to migrate GSM-R, a radio standard based on 2G technology to LTE or 5G, which would use radio or microwave technology to transmit voice/data and avoid the use of complex and expensive cables. Other components that are part of the system are being developed at the same time, such as EVC (European Vital Computer), MDI (Machine Driver Interface), JRU (Juridical Recorder Unit), RBC (Radio Block Center - ETCS Centrale), etc. Existing technologies enable ATO process-Automated Train Operation, OC (Object reCognition), Satellite Localization, BDA-Big Data Analysis, AI-Artificial Intelligence, CT-Cloud Computing, DM-Digital (Predictive) Maintenance, etc. All these technologies are being developed rapidly and integrated into existing solutions offered by different vendors. On the other hand, there is a need for passengers to be informed in a timely manner about the possible delays of trains, to enable them to buy online tickets, to book a possible connection or change the means of transport, and to book cars/bikes that will be available to them upon completion of their travel.

The process of digitization itself, in general, can be seen as the interaction of the following levers, which are also the drivers of this process:

- Digital data (digitalization),
- Connectivity (P2P, M2M, P2M, etc.),
- Automation and
- Digital Customer Access.

Digital Data (digitalization), involves capturing, storing, processing and analyzing digital data, allows better predictions and decisions to be made. Connectivity means interconnection between all participants, via mobile, radio, WiFi or fixed-line. Automation is a combination of traditional technology and new AI technology, that enables system to work autonomously and self-organize. Digital Customer Access means that the Internet provides direct access to any kind of services.[3]

B. Risks of the digitalization process

As already emphasized, the process of digitization carries with certain risks, which grow exponentially with the introduction of new services and CotS components. As can be seen from Figure 1, until the early 2010s, there was no great risk of cyber-attack, due to the closure of the rail system and the traditional approach. Until then, AT&T Hack attacks were in progress, Melissa Worm, Morris Worm, and the first DDoS attacks to occur. Shortly thereafter, the ILOVEYOU virus (known as Loveletter) appeared.[4] Railways and industrial systems were not seriously threatened by their closeness to the Internet and external networks. More serious threat to the railway, as well as any system based on ICS (Industrial Control System) and SCADA (Supervisory Control and Data Acquisition) systems, started with Stuxnet (attack on Siemens SCADA system) and later security omission in OpenSSL that exploited the Heartbleed-bug.[5] 2016 was marked by WannaCry, which endangered several dozen European large companies, including the German DB. It was a crypto-attack of ransomware.[6] During 2017, ESET analyzed Stuxnet and rated this malware as very dangerous because it could interfere with critical infrastructure operations.[7]

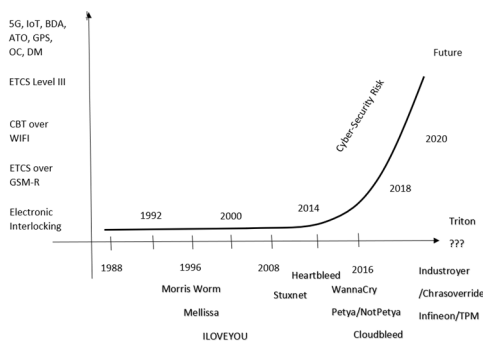


Figure 1 Cyber-security risk function

Figure 1 shows the exponential growth of cyber-security risk by introducing new services and technologies such as 5G, IoT, BDA (Big Data Analysis), ATO (Automatic Train Operation), GPS, OC (Object) reCognition), DM (Digital

Maintenance), etc. That everything is not "sci-fi", but reality is shown by concrete data and recorded attacks in the railway domain. According to [8] the 14-year-old boy from Poland modified a TV remote control to use it to change Tram track points. In 2012, the Transportation Security Administration (TSA) reported that hackers manipulated a railway computer on a Northwest rail company. Additionally, the Britain's rail network has been hit by at least four major cyber-attacks last year as the nation remains under threat from devious Islamic State terror hackers. [9] The latest WannaCry ransomware, which has affected over 200,000 computers worldwide, has not bypassed the railways, and has also affected one small part of the information system of DB (Deutsche Bahn), the largest railway operator in Germany and one of the largest in the world. In 2019, according to [10] SIXGILL, one hacker group on the Dark Web was selling admin access to the admin panel of a Chinese Rail Control System. To show that it was not a "fake", hackers installed as many as four print-screens of management systems, which could see system configurations information about access to specific modules, and even the codes of individual segments of the locomotive configuration.[10]

III. CHALLENGES

There are increasing challenges and open questions that plague cyber-security experts in this domain, both the phase of cyber-attack detection and incident response and digital forensics. A benchmark in the transportation sector from 2017 shows overall cyber maturity as low (average 1.5 on a scale of 5). There are many reasons for this, and some of them have already been mentioned in this paper. Digital forensic analysis of the railway system has not been vacant so far, for the simple reason that until the last 2-3 years, due to the characteristics of the system itself, there were no serious attacks that needed to be analyzed (except for the ones already listed). The complexity that digitalization brings with it and the introduction of new services certainly brings new challenges with certain questions, which is difficult to answer today:

- Do we know the system – all components in the system?
- Do we have good technical documentation of system (there is a lot of vendors, stakeholders)?
- Can we (do we know how?) write a good plan for penetration test?
- Is software we use “good enough”?
- How can we make a forensic analysis of the “black-box” we have (a lot of vendor- specific IoT)?
- How to analyze all data, when we do not know where they are (somewhere in the Cloud)?
- How to extract data from “black box” and safety-relevant systems while avoiding security risks?
- Does the gathered “evidence” have assured integrity?
- Do we have good recovery strategy and downtime procedure which will ensure, that we have a system as it was before an incident and which can be trusted again?
- Are results of analysis “correct” and how can we compare and measure these results?

- How can this analysis be performed with only a minimum or zero impact on setting the system back to an operational mode?

Digital forensic analysis in such an environment can be very complex. There are many reasons for this:

- a large number of components,
- the specificity of them,
- the closeness, and the variety of architectures and hardware/software.

There is no model for digital forensic investigations in the railway domain, but some authors [11] offer a model that is flexible and can be applied. Digital investigation process sometimes involves the analysis of standard ICT components, a PC running Windows/Linux, or some standard hardware component that may possibly contain digital evidence. On the other hand, it is sometimes necessary to collect data or to know the status of an ICS (Industrial Control System) or SCADA (Supervisory Control and Data Acquisition) component of the system. In any ICS, forensic investigation is very difficult task. PLC and DCS systems are embedded systems with vendor dependent software, proprietary languages, architecture, dedicated hardware, and vendor-dependent protocols are in use. Sometimes, without the help of the manufacturer and engineers working on a specific system, getting any data/information will not be possible.



Figure 2 BPMN digital investigation model

Very few authors (scientists) address this issue while the problems of data acquisition from industrial systems themselves have been emphasized by the authors.[12-17]

In some situations, it is necessary to do the so-called “chip-off analysis”, which further complicates the situation. The new ETCS system also brings some technical solutions that

bring a new dimension to forensics in the rail domain- for example, the JRU (Juridical Recording Unit), a device that is actually a "train event recorder" and is intended for collecting, storing and subsequently using data from vital parts of the train, including audio and video signals.[18]

In Figure 2 is shown Business Process Model and notation diagram, which describe this digital investigation process with specificities in railway domain. As we can see, for any investigation we need to have an allowance. This allowance needs to be obtained not only from CISO and board members but also from work council. Data protection department must also be informed in every situation. From another side, when critical infrastructure is hit, LEAs (Law Enforcement Agency) and State Agency for Information Security must be informed. Search warrant must also be obtained in writing. As we already have seen, in the railway environment, we always have vendor-dependent systems, and without help of engineers and experts from this company, collection and examination of evidence will be not possible. In forensic analysis sometimes is necessary presence and engagement of court expert witness (court decision). All the time, throughout the whole process, a chain of custody must be ensured and maintained, because the integrity of digital evidence must be ensured.

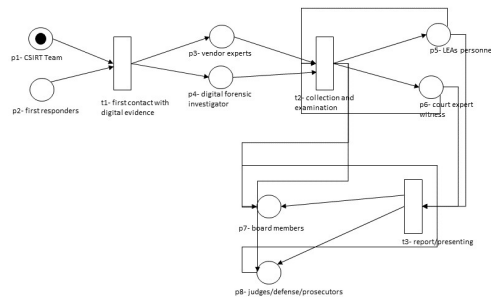


Figure 3 Petri net model showing impact of persons on digital evidence

From another side, there are a lot of stakeholders and persons, who will be involved in this process, which is not so good, because digital evidence is “just a digital data”. That means that digital evidence is very sensitive, can easily be changed, deleted, moved- simply can “disappear”. Personnel who may come into contact with digital evidence are: first responders, forensic investigators, court expert witness, law enforcement personnel, police officer (crime inspectors), CSI personnel, vendor- engineers, specific domain experts, victim, suspect, passerby, judges, prosecutors, board members. In some specific situation additional staff can extend this list.[19]

Considering, numbers of stakeholders, personnel who will be involved, system complexity, life cycle of digital evidence, another aspects and variables (collection time, collection place, collection method, collection reason, etc.), it can be concluded, that ensuring integrity and chain of custody of digital evidence, in environment like this, is very complex. In each of the stages, in this digital investigation process, this “piece of data” is exposed to external influences.

This complexity are shown in Figure 3, where this impact is modeled with Petri-nets. With t-transition is presented a phase in digital forensic investigation model (first contact,

collection and examination- analysis, reporting, etc.). With p-place is presented where digital evidence is located, and arrows are used to link places and transition. Digital evidence is presented like a token which traverses from one location to another.[20]

IV. CONCLUSION

In this paper, authors have dealt with new upcoming technology and the challenges forensic experts face in digital forensic investigations, which will occur more and more in the future. Attention was drawn to open-ended questions and unanswered questions, to open a discussion and possibly encourage more scientist and experts to address this issue. The specificity of the critical infrastructure and complexity of dealing with digital forensics in such an environment is emphasized. A specific BPMN process and human impact on digital evidence, modeled with Petri nets are also given.

ACKNOWLEDGMENT

This paper arose as a part of activity carried out in last few months at DB Netz, to build a good framework and model for a modern, effective Team, which can respond to all challenges in railway domain in this dangerous digital age.

Research supported in part by EC H2020 5G-VICTORI GA 857201

Through research and writing this paper, authors had a selfless help of our Team member, Cyber Security Expert - Alexander Thiel. We want to thank him for reading this paper and giving good advice from hacker's perspective.

REFERENCES

- [1] C. Schlehuber and M. Heinrich, "Challenges and Approaches in Securing Safety - Relevant Railway Signalling," in IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, 2017.
- [2] H. Leister, "Zukunftswerkstatt Schienenverkehr - ETCS und digitale Technologie für Stellwerke," Eisenbahn-Revue International 8-9/2017, pp. 417-422, 2017.
- [3] R. Berger, "The digital transformation of industry", BDI, 2015
- [4] N. Potter, "Top 10 Computer Viruses and Worms", ABC News, available (<https://abcnews.go.com/Technology/top-computer-viruses-worms-internet-history/story?id=8480794>)
- [5] Z. Yhang, Q. Liu, Z. Liu, "A Snort-based Approach for Heartbleed Bug Detection", Proceedings of the 2014 International Conference on Computer Science and Electronic Technology, 2014
- [6] S. Mohurle, M. Patil, "A brief study of WannaCry Threat: Ransomware Attack 2017", International Journal of Advanced Research in Computer Science, Vol.8, No.5, 2017
- [7] A. Matrosov, E. Rodionov, D. Harley, J. Malcho, "Stuxnet Under the Microscope", ESET report, 2018
- [8] Schoolboy hacks into city's tram system, The Telegraph, 2008 (available: <https://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html>)
- [9] Four Cyber Attacks on UK Railways in a Year, Sky news, (available: <https://news.sky.com/story/four-cyber-attacks-on-uk-railways-in-a-year-10498558>)
- [10] Hacked Chinese Rail Control System, SIXGILL Report, 2019, (available: <https://www.hackread.com/wp-content/uploads/2019/03/Hacked-Chinese-Rail-Control-System.pdf>)
- [11] J. Cosic, M. Baca and Z. Cosic, "Chain of Digital Evidence Based Model of Digital Forensic Investigation Process", International Journal of Computer Science and Information Security, Vol.9, No.8, pp. 18-24, 2011.
- [12] P. Van Vliet, M. Kechadi and N.-A. Le-Khac, "Forensics in Industrial Control System: A Case Study," Lecture Notes in Computer Science, pp. 147-156, 2016.
- [13] P. Eden and a. all, "A Forensic Taxonomy of SCADA Systems and Approach to Incident Response," in 3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015), University of Applied Sciences, Ingolstadt, Germany, 2015.
- [14] T. Kilpatrick, T. Gonzalez, R. Chandia, M. Papa and M. Sheno, "An architecture for SCADA network forensics," in International Federation for Information Processing, Volume 222. Advances in Digital Forensics II., Boston, Springer, 2006, pp. pp. 273- 285.
- [15] T. Spyridopoulos and e. al., "Incident Analysis & Digital Forensics in SCADA and Industrial Control Systems," in System Safety Conference incorporating the Cyber Security Conference 2013, 2013.
- [16] P. N. Taveras, "SCADA Live Forensic: Real Time Data Acquisition Process to Detect, Prevent or Evaluate Critical Situation," in 1st Annual International Interdisciplinary Conference, AIIC 2013, Azores, Portugal, 2013.
- [17] J. Stirland and a. all, "Developing Cyber Forensics for SCADA Industrial Control Systems," in Proceedings of the International Conference on Information Security and Cyber Forensics, Kuala Terengganu, Malaysia., 2014
- [18] ERTMS - Subsystems and Constituents of the ERTMS, Subsystems and Constituents of the ERTMS, European Commission, Mobility and Transport, 2019
- [19] J. Cosic, M. Baca, "Do we have full control over integrity in digital evidence life cycle", Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces, pp.429-434, IEEE, 2010
- [20] C. Girault, R. Valk, "Petri Nets for System Engineering: A Guide to Modeling, Verification and Application", Springer-Verlag, 2003

SSVDAG* : Efficient Volume Data Representation Using Enhanced Symmetry-Aware Sparse Voxel Directed Acyclic Graph

Peter Čerešník

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice
peter.ceresnik@tuke.sk

Branislav Madoš

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice
branislav.mados@tuke.sk

Anton Baláž

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice
anton.balaz@tuke.sk

Zuzana Bilanová

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice
zuzana.bilanova@tuke.sk

Abstract— Volume data, which are mostly formed as the regular grids of scalar or vector values, can represent significant amount of data, when for instance $16K^3$ grid of 8b/vox dataset consumes 4TB of space not only on secondary storage, but also in operating memory of computer or graphics card. That is why compact representations of volume data, based on lossless compression, are designed. Family of data structures based on octrees was developed, including Sparse Voxel Octree (SVO), Sparse Voxel Directed Acyclic Graph (SVDAG) and Symmetry-aware Sparse Voxel Octree (SSVDAG). The paper introduces SSV DAG* data structure, which represents further enhancement of the SSV DAG, and provides better compression results without significant increase of decompression overhead.

Keywords—voxelization, volume data, sparse voxel octree, SVO, sparse voxel directed acyclic graph, SVDAG, symmetry-aware sparse voxel directed acyclic graph, SSV DAG, Morton order

I. INTRODUCTION

Original motivation for volumetric data representation was the requirement of visualization of scientific, medical and industrial data. Volume data are mostly represented as the spatially uniform, regular grids of scalar or vector values and are obtained by specialized equipment or as the result of different kinds of simulations. In medical imaging the source of data can be Computed (axial) Tomography (CT), MicroCT or Magnetic Resonance Imaging (MRI). In industry it can be industrial Computed Tomography, which can be source of data for assembly inspection and measurements, flaw detection, failure analysis, reverse engineering and others. As the development of volumetric data visualization continued, the possibilities of its use have also shifted to the field of computer graphics, visual arts and entertainment, including computer games, virtual reality, and photorealistic rendering [1]. Volume rendering is also widely used in the movie industry, with movies such as the XXX, Lord of the Rings, The Day after Tomorrow, Pirates of the Caribbean, the The Mummy 3 etc. [2].

Processing voxelized volumetric datasets, especially if it is to be interactive or real-time, is still a significant challenge,

especially if the size of the data to be processed represents often as large amount that their binary representation cannot be completely stored in the computers or graphics card memory. Therefore, great attention is paid to optimizing the representation of volumetric data in memory by using dedicated data structures for this purpose. Their design often takes into account the requirement for losslessness of the data compression used and assesses not only the requirements of the storage space, but also the time and space complexity of the algorithm designed to compound this data structure from the input voxelized data. Hierarchical data structures such as Sparse Voxel Octrees (SVOs), which were later improved into Sparse Voxel's Directed Acyclic Graph (SVDAG) and later Symmetry-aware Sparse Voxel Directed Acyclic Graph (SSVDAG), have earned great attention in this regard. That is why we decided to contribute to the field of hierarchical data structures for voxelized scene representation, with the modification of SSV DAG data structure.

Contribution of the paper is in modification of SSV DAG data structure that brings increase of the compression ratio in comparison to the SSV DAG.

The structure of this paper is as follows:

Section II deals with the related work covering linearization of multidimensional data using Morton order and hierarchical representations of two and three dimensional datasets. Section III of the paper deals with the detailed outline of Symmetry-aware Sparse Voxel Directed Acyclic Graph (SSVDAG) data structure and in this paper proposed modification of this data structure SSV DAG*. Section IV represents tests that were provided with the use of three dimensional scenes originally stored in geometry definition file format Wavefront OBJ. Volume datasets were obtained by the process of voxelization and then SVO, SVDAG, SSV DAG and SSV DAG* compression was applied. Section V of the paper summarizes conclusions that are based on the above mentioned test results and outlines future research in the field of hierarchical data structures dedicated to volume data representation.

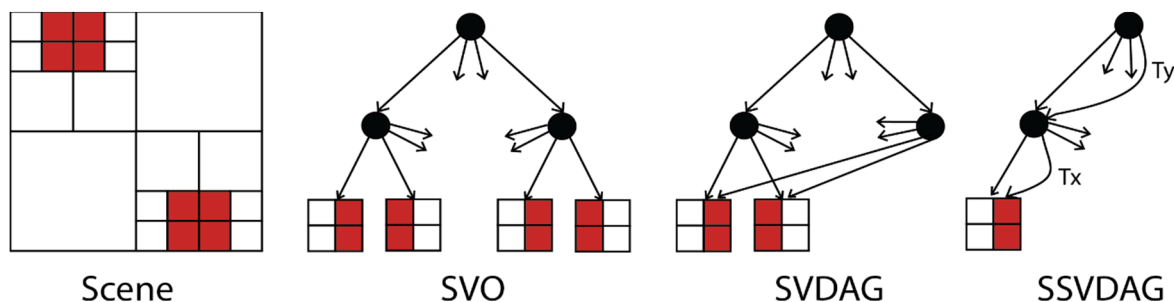


Fig. 2 Original scene and its representation in Sparse Voxel Octree (SVO), which efficiently represents empty subspaces, Sparse Voxel Directed Acyclic Graph (SVDAG) which allows to merge identical subtrees and Symmetry-aware Sparse Voxel Directed Acyclic Graph (SSVDAG) with the possibility to merge subtrees which are identical when similarity transformation is applied.

II. RELATED WORK

The section contains only a selection of papers that are closely related to the solution presented in this paper, due to the large amount of the work published in the field of hierarchical representation of voxelized datasets.

linearization of multidimensional grids - Morton order introduced in [3], referenced also as the Morton sequence, Morton code or Z-order is used for linearized representation of n-dimensional grids of data, using Z-order curve (space filling curve in the shape of the character Z). Morton order is widely used in the representation of voxelized volumetric datasets (Figure 1).

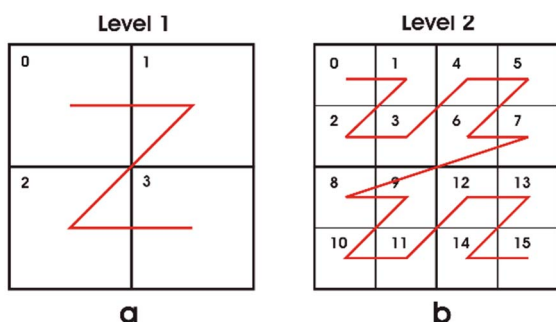


Fig. 1 Morton order, using z-order curve as the tool for linearization of multidimensional grids of data. Example shows two-dimensional data linearized a) to the first level and b) to the second level.

voxelization Out-of-core voxelization algorithm for construction of sparse voxel octrees from triangle meshes was introduced in [4]. The algorithm consists of two steps. First step represents voxelization, in which high resolution 3D grid of voxels is produced from triangles represented objects. Second step transforms intermediate 3D grid of voxels and produces sparse voxel octree. The algorithm allows that not only input triangle mesh, output sparse voxel octree and intermediate high-resolution three-dimensional voxel grid represented in Morton order to be larger than available memory, but it allows to use only a fraction of memory in comparison with in-core algorithms with the use of comparable time.

hierarchical representation of two dimensional data In [5] there was discussion related to the possibility of the use of quadtrees for representation of two-dimensional binary data using Common Subtree Merging (CSM). Besides of the homogeneously empty subtrees, which represent quadrants homogeneously filled with passive voxels, it describes also

usage of subtrees which are homogeneously filled, it means that are representing quadrants homogeneously filled by active pixels. Data structure called CSM-quadtree was designed.

In [6] two-dimensional template-based encoding (2DTE) technique was introduced for the construction of quadtrees for the use in the field of cartography. Lossless compression was based on the merging of common subtrees with the use of template mapping and Morton sequence. This approach was then extended to the three dimensional voxel data by [7].

hierarchical representation of three dimensional data Octrees as the data structure for representation of volume data are used for decades, from the 80's of the last century we can mention [8][9][10]. In [11] there is focus on the use of hierarchical data structures such as the octree to speed up the determination of the objects that are intersected by rays emanating from the viewpoint.

Sparse Voxel Octrees (SVO) were introduced as the data structure based on hierarchical representation of the volume based on the use of octant trees that can represent volume datasets in compact form in which empty subspaces are not represented by full subtrees. Data structure is suitable for archiving purpose, but less suitable for immediate traversing, because of the absence of child pointers.

Efficient Sparse Voxel Octrees (referenced in sources as the ESVO) data structure was introduced in 2010 in [12]. Voxels are enhanced using contour information which helps to raise geometrical resolution and allow compact representation of smooth surfaces and makes the speed of raytracing higher.

There was data structure called Sparse Voxel Directed Acyclic Graph (SVDAG) introduced in [13] in 2013, which transforms SVO into directed acyclic graph. It adds 32-bit pointers to the child nodes and multiple common subtrees are replaced by the representation of one instance of the subtree which is multiply referenced.

In 2016 evolution of SVDAG was created called Symmetry-Aware Sparse Voxel Directed Acyclic Graph (SSVDAG) and it was introduced in [14]. This data structure allowed further compression with minimal influence to the decompression overhead in time of tree traversing. SSVDAG allows merge of subtrees which are identical when similarity reflection transformation is used. Another contribution is the use of frequency-based pointer compaction, which sorts nodes by the frequency of their referencing in each level of the tree separately. Long 32-bit and short 16-bit child node pointers were introduced.

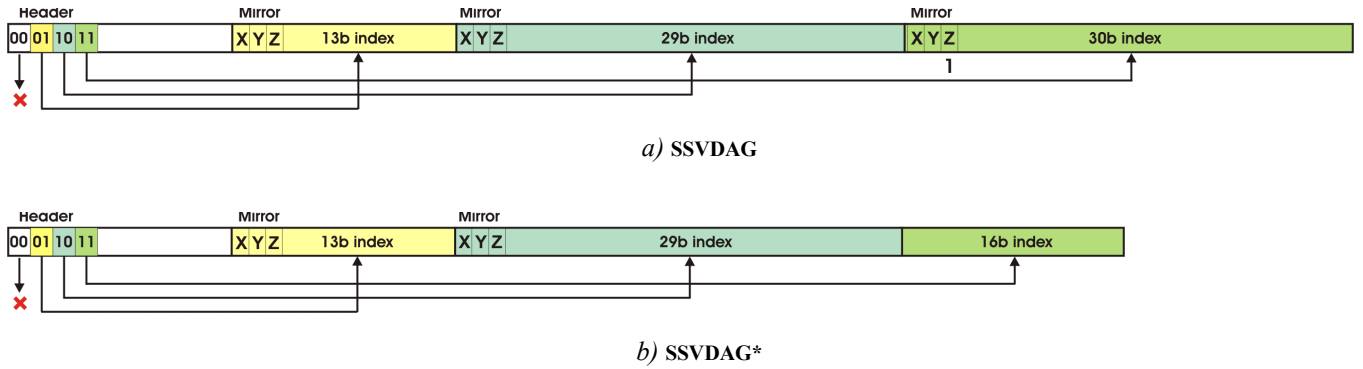


Fig. 3 Symmetry-aware Sparse Voxel Directed Acyclic Graph (SSV DAG) binary representation of inner nodes in a) original form with short, long and extra-long pointers to the child nodes and b) modified SSV DAG* data structure which replaces extra-long pointer by short pointer without encoded symmetry transformation that introduces bigger addressing space of short pointers.

III. MODIFIED SSV DAG ENCODING

Brief introduction of SSV DAG data structure is shown in the next subchapter followed by the description of modified version of this data structure SSV DAG*, which is the contribution of this paper.

A. SSV DAG

Symmetry-Aware Sparse Voxel Directed Acyclic Graph (SSV DAG) is hierarchical data structure that consists of inner nodes which binary representation is shown in the Figure 3a and leaf nodes, with different optimized binary structure represented by three dimensional grids of voxels.

inner nodes are represented by the Header and appropriate number of child pointers.

Header of the node includes 8 Header Tags (HT), when each HT consists of two bits and represents one of eight octants.

HT ‘00’ represents empty octant, it means the octant in which there is no active voxel.

HT ‘01’ represents non-empty octant and 16b child pointer is used to address the appropriate child node. First 3 bits are dedicated to encoding of mirroring in respective axis and another 13 bits are reserved for the address of child node. Addressing space is in that case 16KB for inner nodes (data is 16-bit aligned) and 64KB in the case of leaf nodes (data is 64-bit aligned).

HT ‘10’ represents non-empty octant and 32b child pointer is used. Because first 3 bits are dedicated to mirroring, 29 bits are useful for addressing of child nodes. It creates 1GB of addressing space for inner nodes and 4GB of addressing space in the case of leaf nodes.

HT ‘11’ represents non-empty octant and 32b child pointer is used. First 3 bits are dedicated to the mirroring and another 29 bits are used for addressing as it is in the previous type of child pointer. The difference is, that bit 30 of the address is set to 1 in the address, which adds another 1GB of addressing space for inner nodes and 4GB of leaf nodes.

In each level of the data structure nodes are ordered according to the number of pointers that are pointing to them, from the most targeted nodes to the less targeted.

Most targeted nodes are in the addressing space of short 16b long child pointers, which allows to minimize the overall length of child pointers. Child pointers are representing offsets from addresses of first child nodes in each particular level of nodes in the data structure.

Mirroring, represented by 3 bits of child pointer indicates, that child node is mirrored in the particular axis. This operation is independent in each axis, and that is why there are eight different possibilities of mirroring.

leaf nodes two least levels of nodes are compacted into grids with the resolution of 4^3 voxels each. Each voxel is represented by one bit, when passive voxel is represented by bit 0 and active voxel is represented by the bit 1. This solution brings possibility to eliminate child pointers between penultimate and the last level of nodes.

B. SSV DAG*

Modification of the SSV DAG data structure introduced in this paper brings change to the use of the Header Tag (HT) ‘11’. In original SSV DAG data structure it indicates the use of 30 bit extra-long child pointer (with the most significant bit set to 1) along with the use of another 3 bits for similarity transformation encoding. It allows to expand addressing space in comparison to 29 bits long child pointer with the HT ‘10’.

In SSV DAG* data structure the modification was introduced, in which HT ‘11’ indicates short child pointer which is 16-bits long without the use of similarity encoding. This child pointer can be used in case that no similarity transformation is needed. Addressing space is 8 times larger in comparison with the short child pointer indicated by the use of HT ‘01’, which has only 13bits for encoding of the child address (another 3 bits are used for similarity transformation encoding).

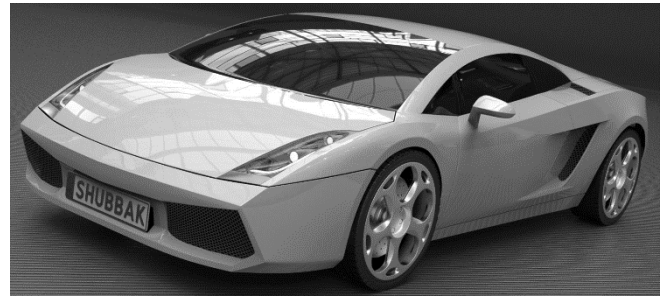
Advantage of this solution is in the expansion of the addressing space of short 16-bit child pointers to 128KB resp. 512KB. Some of long child pointers are transformed into 16 bit child pointers indicated by the use of HT ‘11’. Because in this addressing space child nodes are more frequently addressed, it makes the effect of long child pointers shortening even stronger.

IV. RESULTS AND DISCUSSION

In this section of the paper we evaluate different aspects of solution, we show several results for our SSV DAG* data structure in comparison with the other data structures. Programs were implemented in C/C++ and we have tested their performance by computer with Intel i7 7700K CPU with 16 GB RAM and Nvidia GeForce 1080 8 GB. The scene used for testing was 3D projection of automobile Lamborghini Gallardo, Abandon Warehouse and Great colonnade of Apamea which visualizations can be seen at Fig. 4.

A. Memory consumption

In the Table I we show the number of bits per active voxel for given scene at different resolution and when using different data structures. Our implementation of SSV DAG with modified encoding is referred as SSV DAG*. At lower resolution only short pointers are used and that is why there is no difference between original and modified encoding. In Gallardo scene, starting at 512^3 resolution we can see difference between encodings, which is relatively biggest at moderate resolutions 1024^3 and 2048^3 with maximum of 9.2% of memory consumption reduction when using modified encoding. At high resolutions memory cost of long pointers is in so significant measure, that difference between encoding in bits per voxels is relatively smaller as it can be seen also in the Table II and Figure 5. But there is still notable reduction of used memory. In general it can be said that our modified encoding provides always at least as good memory consumption as SSV DAG encoding, but in most cases it provides additional reduction. When taking into account different structures, SVO has the best bit per voxel ratio at small resolutions, because of the absence of pointers. It makes it good structure for offline storage, but the traversing requires big overhead. Basically it means that pointers must be added before traversing.



a)



b)



c)

Fig. 4 Visualizations of examples of scenes that were used for testing purpose. a) Lamborghini Gallardo b) Abandoned Warehouse c) Great colonnade of Apamea

Table I Bits per active voxel according to different resolutions and data structures.

Structure	Resolution							
	128^3	256^3	512^3	1024^3	2048^3	4096^3	8192^3	16384^3
Lamborghini Gallardo								
SVO	2.441	2.537	2.756	2.882	3.009	3.157	3.243	3.287
SVDAG	10.254	8.773	9.411	8.400	7.382	5.948	4.540	3.453
SSVDAG	3.504	3.099	3.188	3.225	2.983	2.692	2.271	1.912
SSVDAG*	3.504	3.099	3.128	2.928	2.728	2.596	2.224	1.880
Abandoned Warehouse								
SVO	3.740	4.938	6.126	5.809	4.755	4.001	3.673	3.619
SVDAG	20.689	25.682	30.672	26.280	21.438	17.436	14.686	11.846
SSVDAG	9.197	10.888	13.416	11.156	9.215	7.709	6.519	5.220
SSVDAG*	9.197	10.888	13.416	11.156	8.844	6.776	5.598	4.812
Great colonnade of Apamea								
SVO	2.730	2.870	3.144	3.370	3.496	3.575	3.630	3.663
SVDAG	11.959	12.056	12.566	12.226	10.804	8.792	6.795	5.344
SSVDAG	4.399	4.348	4.976	5.107	4.594	3.934	3.359	2.953
SSVDAG*	4.399	4.348	4.503	4.465	4.301	3.824	3.297	2.905

SVDAG which uses pointers with fixed size is ineffective at low resolutions, but with increasing resolutions it is comparable with SVO, in which inability to reference node more than once makes lower levels of the data structure voluminous. Traversing of SVDAG brings benefits in lower overhead in comparison to SVO. SSV DAG shares good bit per voxel ratio through all tested resolutions. Starting with resolution 2048^3 it provides best ratio from all given data structures. Encoding with more pointer sizes brings better ratio than SVDAG has. Additional compression results also from using grids for encapsulating two lowest levels and carving out mutually symmetric subtrees. SSV DAG can be effectively traversed with small overhead, which makes it better structure for this purpose than SVO.

B. Construction speed

In this test we provide the construction speed measurements of compared four data structures as it can be seen in Table II. SVO was constructed out-of-core from the stream of active voxels and stored without pointers. SVDAG was also constructed out-of-core, processing two levels at the same time, the same applies for SSV DAG. SVDAG was created by processing SVO and SSV DAG by processing SVDAG with grids at the lowest level. Construction of SVO is in the most cases the fastest, because there are no extensive sortings or comparisons. Despite that at the highest tested resolution SVO construction is not the fastest because for given scene there are more than 300 million active voxels that need to be processed. Construction of SVDAG tends to be slower at higher resolution in comparison to other two structures because there is enormous number of nodes at the two lowest levels.

Table II Memory consumption comparison for different resolutions in voxels between SSV DAG* and SSV DAG, when SSV DAG means 100% and value in table = $SSV DAG^* / SSV DAG * 100$ [%].

Resolution	Lamborghini Gallardo	Abandoned Warehouse	Great colonnade of Apamea
128^3	100,0%	100,0%	100,0%
256^3	100,0%	100,0%	100,0%
512^3	98,1%	100,0%	90,5%
1024^3	90,8%	100,0%	87,4%
2048^3	91,5%	96,0%	93,6%
4096^3	96,4%	87,9%	97,2%
8192^3	97,9%	85,9%	98,2%
16384^3	98,3%	92,2%	98,4%

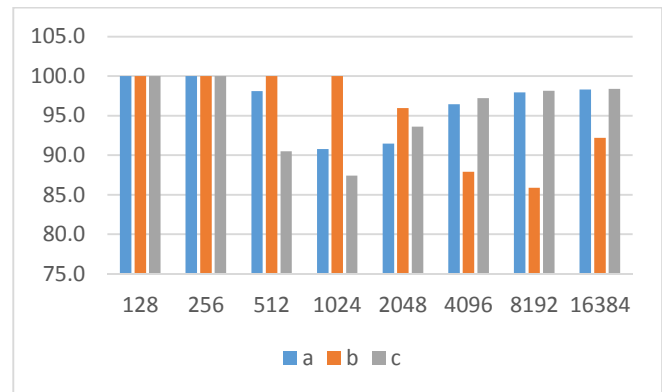


Fig. 5. Memory consumption comparison for different resolutions in voxels between SSV DAG* and SSV DAG, when SSV DAG means 100% and value in table = $SSV DAG^* / SSV DAG * 100$ [%] for a) Lamborghini Gallardo b) Abandoned Warehouse and c) Great colonnade of Apamea.

Table III Construction speed in seconds according to different resolutions and data structures.

Structure	Resolution							
	128^3	256^3	512^3	1024^3	2048^3	4096^3	8192^3	16384^3
Gallardo								
SVO	0.021s	0.078s	0.381s	1.476s	5.855s	26.604s	104.606s	409.851s
SVDAG	0.019s	0.080s	0.382s	1.632s	7.034s	30.702s	133.252s	554.580s
SSV DAG	0.040s	0.139s	0.591s	2.122s	8.086s	29.883s	106.351s	380.716s
SSV DAG*	0.040s	0.138s	0.588s	2.109s	8.062s	29.734s	105.926s	378.812s
Abandoned Warehouse								
SVO	0.002s	0.005s	0.010s	0.028s	0.082s	0.319s	1.312s	5.176s
SVDAG	0.005s	0.014s	0.037s	0.106s	0.292s	0.735s	2.202s	7.838s
SSV DAG	0.011s	0.032s	0.082s	0.245s	0.567s	1.325s	3.571s	10.639s
SSV DAG*	0.011s	0.032s	0.082s	0.244s	0.562s	1.322s	3.553s	10.596s
Great colonnade of Apamea								
SVO	0.028s	0.109s	0.489s	1.577s	5.890s	24.463s	89.488s	343.788s
SVDAG	0.029s	0.132s	0.567s	2.037s	8.433s	35.182s	141.363s	589.046s
SSV DAG	0.056s	0.220s	0.889s	3.170s	11.578s	41.723s	147.061s	490.819s
SSV DAG*	0.056s	0.218s	0.884s	3.154s	11.532s	41.640s	146.326s	488.856s

Compression of two lowest levels into grids provides speed up for SSVVDAG construction that is notable mainly at higher resolution. SSVVDAG construction is comparable with SVO construction even there are more sortings and related operations than while constructing SVDAG. Number of grids in SSVVDAG is much lower than would be number of corresponding nodes and sorting and processing of grids is less complex in general.

Construction speed of the SSVVDAG* data structure is fraction of the time quicker that it is in the case of the unmodified SSVVDAG data structure.

V. CONCLUSION

The paper deals with the problematics of hierarchical data structures which are dedicated to losslessly compressed representation of the geometry of voxelized volumetric datasets with the aim to build on previous development in this area, represented above all by Sparse Voxel Octrees (SVO), Sparse Voxel Directed Acyclic Graph (SVDAG) and Symmetry-aware Sparse Voxel Directed Acyclic Graph (SSVDAG).

The contribution of the paper is in the design of modified SSVVDAG data structure called SSVVDAG*, which brings, according to provided tests, at least the same compression or better compression results in comparison to original SSVVDAG data structure in most cases without significant increase of the decompression overhead in time of data structure traversing and with the same construction speed as the SSVVDAG data structure. The best gains in compression can be seen in scenes which are not able to use only short pointers and with the extensive use of long pointers the compression gains are lowering.

In the future research we will focus our attention to the possibility of further compression of the length of child pointers and we will evaluate the possibility to use developed data structure in other fields than computer graphics, like data flow computer architectures [15], source code analysis [16][17] and computer security [18][19] where we will try to include granularity into the design of the data structure to obtain further compression gains and make the structure more suitable for different application domains.

ACKNOWLEDGMENT

This work was supported by KEGA Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under Grant No. 003TUKE-4/2017 Implementation of Modern Methods and Education Forms in the Area of Security of Information and Communication Technologies towards Requirements of Labor Market and under Grant No. 077TUKE-4/2015 Promoting the interconnection of Computer and Software Engineering using the KPIkit.

REFERENCES

- [1] L. Vokorokos, E. Danková, and N. Ádám, „Task scheduling in distributed system for photorealistic rendering“, In Proc. Of the IEEE 8th International Symposium on Applied Machine Intelligence and Informatics (2010), pp. 43-47, 2010.
- [2] C. Crassin, F. Neyret, S. Lefebvre and E. Eismann, “Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering”. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D). ACM, ACM Press, Boston, MA, Etats-Unis.
- [3] G. Morton, “A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing”, Research Report. International Business Machines Corporation (IBM), Ottawa, Canada, 20.
- [4] J. Baert, A. Lagae and P. Dutré, “Out-of-core construction of sparse voxel octrees”. In Proceedings of the 5th High-Performance Graphics Conference. HPG '13. ACM, New York, NY, USA, 27–32.
- [5] R.E. Webber and M. B. Dillencourt, “Compressing quadtrees via common subtree merging”. Pattern Recognition Letters 9 (1989) , April 1989, pp. 193–200.
- [6] H. Ker-Chang Chang, L. Shing-Hua, and T. Cheng-Kuan, “Two dimensional template-based encoding for linear quadtree representation”
- [7] E. Parker and T. Udeshi, “Exploiting self-similarity in geometry for voxel based solid modeling”. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications. SM '03. ACM, New York, NY, USA, 157–166.
- [8] S. M. Rubin and T. Whitted, “A 3-dimensional representation for fast rendering of complex scenes”. In SIGGRAPH Comput. Graph. ACM, New York, NY, USA, 110–116.
- [9] C. Jackins and S. Tanimoto, “Octrees and their use in representing three-dimensional objects”. Computer Graphics and Image Processing 14, 3, 249–270.
- [10] D. Meagher, “Geometric modeling using octree encoding”. Computer Graphics and Image Processing 19, 2, 129–147.
- [11] H. Samet, “Implementing raytracing with octrees and neighbor finding”. Computers And Graphics 13, 445–460. SAMET, H.1989. Implementing raytracing with octrees and neighbor finding. Computers And Graphics 13, 445–460.
- [12] S. Laine and T. Karras, “Efficient sparse voxel octrees”. In Proceedings of ACM SIGGRAPH 2010 Symposium on Interactive 3D Graphics and Games. ACM Press, New York, NY, USA, 1–9.
- [13] V. Kämpe, E. Sintorin and U. Assarsson, “High resolution sparse voxel DAGs”. ACM Trans. Graph. 32, 4, Article 101 (July 2013), 13 pages. DOI: <https://doi.org/10.1145/2461912.2462024>.
- [14] A. J. Villanueva, F. Marton and E. Gobbetti, “SSVDAGs: symmetry-aware sparse voxel DAGs”. In Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '16). ACM, New York, NY, USA, 7-14. DOI: <https://doi.org/10.1145/2856400.2856420>.
- [15] N. Ádám, “Single input operators of the DF KPI system“, Acta Polytechnica Hungarica, 7(1), 73-86 (2010)
- [16] J. Juhár and L. Vokorokos, “Separation of concerns and concern granularitz in source code. Informatics 2015. – Danvers : IEEE, 139 – 144 (2015).
- [17] Šimoňák, S. „Algorithm visualizations as a way of increasing the quality in computer science education“, In: SAMI 2016. - Danvers : IEEE, 2016 P. 153-157. - ISBN 978-146738740-8.
- [18] L. Vokorokos, A. Pekár, N. Ádám and P. Daranyi, „Yet Another Attempt in User Authentication, 2013, vol. 10., issue 3, pp. 37-50, Acta Polytechnica Hungarica (2013).
- [19] L. Vokorokos, M. Ennert, M. Čajkovský and J. Radušovský, „A Survey of parallel intrusion detection on graphical processors. In: Central European Journal of Computer Science, vol. 4, issue 4, pp. 222-230, Open Computer Science, 2014.

Two classes of graphs with symmetries allowing a compact representation

Uroš Čibej

*Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
jurij.mihelic@fri.uni-lj.si*

Jurij Mihelič

*Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
uros.cibej@fri.uni-lj.si*

Abstract—Symmetries are a convenient way to describe redundancies in data. This article presents two novel classes of graphs which can be represented in a compressed format using symmetries. We denote the first class as symmetry-compressible graphs where global automorphisms are used to create the representation. Since automorphisms are scarce in real graphs, we extend this notion by defining near symmetry-compressible graphs, which contain a much larger class of graphs, including many graphs arising in practice. In order to demonstrate the practical potential of the presented classes, we design two compression algorithms and evaluate them on a set of realistic networks.

Index Terms—IEEE, IEEEtran, journal, L^AT_EX, paper, template.

Graphs are a general structure for data representation and are recently being more extensively used, in order to represent complex relations emerging in modern applications. And where there is a lot of data, there is the need for new methods for data compression. Graph symmetries are mathematically defined as graph automorphisms, and we will use these two terms as synonyms. In this paper we introduce a representation of graphs which uses one of the graph automorphisms to potentially reduce the representation size. We define a class of graphs that are compressed using such a representation. Furthermore, we extend the representation to include a much larger class of graphs. The first definition of these concepts was done by the authors in [1].

Even though we start with a rather theoretical definitions in graph theory, we strive for practical applications of the introduced concepts. The second part of the paper is thus dedicated to practical algorithms and an empirical evaluation, which shows the potential for compression of real graph data, as well as data that is not directly given as a graph (such as images). There have been many developed methods for graph compression, see e.g. [2], [3], [4], but to our knowledge none of the methods uses symmetries as the underlying compression mechanisms.

The paper is structured as follows. The next section gives some preliminaries and introduces the concept of symmetry-compressible graphs. The third section extends this concept with near symmetry-compressible graphs. The fourth section starts the practical part of the paper, describing two heuristic algorithms for graph compression, which are finally also empirically analyzed.

I. SYMMETRY-COMPRESSIBLE GRAPHS

In this paper, a graph is presented as a set of edges, $G \subseteq V \times V$, where V is some set of vertices. We will use mostly the set of vertices $\{1, 2, \dots, n\}$. Without loss of generality, we assume the graph has no isolated vertices. To explicitly denote the set of vertices of the graph, we will write $V(G)$.

We work with two related representations of symmetries. The standard representation are permutations of the vertex set, i.e., a bijective function $\pi : V(G) \rightarrow V(G)$ that preserves connectivity $((u, v) \in G \implies (\pi(u), \pi(v)) \in G)$. All such permutations form a permutation group $Aut(G)$. We also require an alternative representation of an automorphism π , namely, the permutation of the edges. The edge permutation, induced by the vertex permutation π , is defined as $\bar{\pi}((u, v)) = (\pi(u), \pi(v))$.

Permutations are considered in a standard cycle notation, i.e., as a set of disjoint cycles. This set contains also the cycles with only one element (identities of the permutation). The notation $cyc(\pi)$ denotes the set of all the cycles of the permutation π . Same goes for the edge permutation $\bar{\pi}$. To obtain the cycle which contains a vertex v (or edge e), we use $cyc(\pi, v)$ ($cyc(\bar{\pi}, e)$ for edges).

Since graphs are represented as a set of pairs, in order to have a comparable representation, also the permutations will be represented as a set of pairs. Trivially, since it is a bijection, a symmetry can be represented by $|V(G)|$ pairs. However we can remove many redundancies in such a representation. Namely, the identities can be omitted, and one pair from each cycle can also be left out. The size (of the representation) is expressed in terms of the cycle sizes as:

$$|\pi| = \sum_{c \in cyc(\pi)} (|c| - 1).$$

A. Definition

Knowing a symmetry π of the graph G , we do not have to represent all the edges of the graph, since some edges can be generated using π . More specifically, for each cycle of the edge permutation $\bar{\pi}$, only one edge is required.

We denote the required part of the graph (under the symmetry π) as G^π , and define it as

$$G^\pi = \left\{ e \in G \mid e = \min_{f \in \text{cyc}(\bar{\pi}, e)} f \right\}$$

and call it a *residual graph*. This definition chooses the minimal element from each cycle of the permutation $\bar{\pi}$. The ordering of the elements can be arbitrary, meaning basically that exactly one element from each cycle is selected.

The pair (π, G^π) is an alternative representation of the graph G , since the entire edge set of the graph can be uniquely reconstructed from it. The reconstruction simply takes every edge of G^π and iteratively applies π to both ends of the current edge, until one cycle of $\bar{\pi}$ is obtained.

The size of this alternative representation is in general different from the size of the original graph, depending both on the graph itself as well as on the symmetry in consideration. We define a family of graphs, which possess symmetries for which the alternative representation is smaller than G .

Definition 1 (Symmetry-compressible graphs). *A graph G is symmetry compressible ($G \in \mathcal{SC}$) if*

$$\exists \pi \in \text{Aut}(G) : |\pi| + |G^\pi| < |G|.$$

Since a graph can have more than one symmetry with the above property, we also define a natural optimization problem. Let us first define two measures of ‘‘compressibility’’, which enables us to compare different symmetries. The first measure is the absolute efficiency of the symmetry on G

$$\Delta(\pi, G) = |G| - |\pi| - |G^\pi|.$$

In order to compare the efficiencies of the symmetry compression on different graphs, we also consider the relative efficiency of the compression, i.e.,

$$\Delta^r(\pi, G) = \frac{\Delta(\pi, G)}{|G|}.$$

For symmetry-compressible graphs $\Delta^r(\pi, G) \in (0, 1)$. Based on these measures, the definition of the optimization problem is the following.

Definition 2 (MaxSC). *Given a graph G find an automorphism $\pi \in \text{Aut}(G)$ which maximizes the efficiency $\Delta(\pi, G)$.*

Notice that the same symmetry maximizes both the absolute and the relative efficiency.

To clarify the introduced concepts, Table I shows a simple graph C_4 , and three of its representative symmetries. Two of these symmetries increase the size of representation, whereas one symmetry reduces the size by 1, meaning $C_4 \in \mathcal{SC}$.

B. Properties

With this simple example we see that symmetry-compressible graphs exist. We now focus on two families of graphs and demonstrate they are not symmetry-compressible. The first family are trees and the second family are cycle

graphs C_k for which we show that compressibility depends on k .

Theorem 1 (*Tree $\notin \mathcal{SC}$*). *If the graph T is a tree,*

$$\forall \pi \in \text{Aut}(T) : |\pi| + |T^\pi| \geq |T|.$$

Proof. A non-trivial (i.e. not an identity) symmetry π in a tree T maps a subtree T_1 into another subtree T_2 . Without loss of generality, we focus only on the subtree $T_1 \cup T_2$, since the part of the tree that remains fixed under the symmetry, remains also in T^π , and can thus be ignored. The size of the representation for the relevant part of the tree is $|T_1 \cup T_2|$.

Notice that there are only two possibilities for T_1 and T_2 : either (1) $T_1 \neq T_2$ or (2) T_1 and T_2 share one common edge, which serves as a pivot for the symmetry. In case of (1) the two trees are edge disjoint, thus $|T_1 \cup T_2| = |T_1| + |T_2| = 2|T_1|$. There are $|T_1| + 1$ vertices in $|T_1|$, but since there is one vertex that remains fixed under π , we need $|T_1|$ pairs to describe the symmetry. The size of the representation is thus $|(T_1 \cup T_2)^\pi| + |\pi| = |T_1| + |T_1| = 2|T_1|$, which is the same size as $|T_1 \cup T_2|$.

In case of (2), one edge remains fixed by the symmetry. The size of $|T_1 \cup T_2| = |T_1| + |T_2| - 1 = 2|T_1| - 1$. The size of the symmetry is again $|\pi| = |T_1|$, since the two two endpoints of the fixed edge can be given with only one pair. The entire representation has size $|\pi| + |(T_1 \cup T_2)^\pi| = |T_1| + |T_1| = 2|T_1|$, which is larger than $|T_1 \cup T_2|$.

In both cases the symmetry representation size is larger or equal to the edge representation, thus no tree is in \mathcal{SC} . \square

Theorem 2. *For cycle graphs C_k it holds: if k is even, $C_k \in \mathcal{SC}$, if k is odd $C_k \notin \mathcal{SC}$.*

Proof. Cycles have two types of symmetries: rotational symmetry and reflective symmetry. It is easy to see that the rotational symmetry does not compress any C_k . Let us suppose we have a rotation for m positions:

$$\begin{aligned} m \mid k \quad |\pi| + |C_k^\pi| &= m \left(\frac{k}{m} - 1 \right) + m = k = |C_k|, \\ m \nmid k \quad |\pi| + |C_k^\pi| &= (k - 1) + 1 = k = |C_k|. \end{aligned}$$

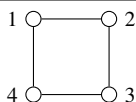
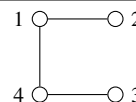
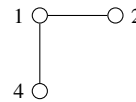

So we focus on reflective symmetry π . For even cycles two options are possible, either the cycle is reflected over two edges, or over two vertices. If it is reflected over two edges, then the symmetry constitutes of $\frac{k}{2}$ pairs. the size of the residual graph $|G^\pi| = \frac{k-2}{2} + 2 = \frac{k}{2} + 1$. making the size of the entire representation $|\pi| + |G^\pi| = \left(\frac{k}{2} + 1\right) + \left(\frac{k}{2}\right) = k + 1 \not< k$. So this type of symmetry does not compress even cycles.

If, on the other hand, it is reflected over two nodes then the two nodes remain fixed. The other $k - 2$ are mapped unto each other, i.e., $|\pi| = \frac{k-2}{2}$. The graph G^π constitutes exactly half of the edges of the original graph, i.e., $\frac{k}{2}$. The entire size of the representation is $|\pi| + |G^\pi| = \frac{k}{2} + \frac{k-2}{2} = k - 1 < k = |G|$. Even cycles are therefore symmetry-compressible.

For odd cycles only one scenario is possible, i.e., the cycle is reflected over one node and one edge (both remain fixed by the symmetry). Thus $|\pi| = \frac{k-1}{2}$. This symmetry maps one edge to itself, all the other edges are mapped to a different edge. This makes the size of $|G^\pi| = \frac{k-1}{2} + 1$. The entire size

TABLE I

THREE REPRESENTATIVE SYMMETRIES π ($\bar{\pi}$ ALSO SHOWN FOR EASIER INTERPRETATION OF RESULTS) OF THE 4-CYCLE, THE CORRESPONDING COMPRESSED GRAPH G^π , AND THE SIZE OF (π, G^π) . THE SYMMETRY SHOWN IN ROW 2 RESULTS IN A SMALLER REPRESENTATION THAN THE ORIGINAL GRAPH, MAKING $C_4 \in \mathcal{SC}$.

G	π	$\bar{\pi}$	G^π	$ \pi + G^\pi $
	(12)(43)	((1, 4)(2, 3))		$2 + 3 \not< G $
	(13)	((1, 2)(2, 3))((1, 4)(3, 4))		$1 + 2 < G $
	(1234)	((1, 4)(1, 2)(2, 3)(3, 4))		$3 + 1 \not< G $

of the representation is $|\pi| + |G^\pi| = \frac{k-1}{2} + \frac{k-1}{2} + 1 = k \not< k$. Odd cycles are therefore not symmetry-compressible. \square

The following theorem demonstrates the relation between π and $\bar{\pi}$ for symmetry-compressible graphs.

Theorem 3. $G \in \mathcal{SC} \iff \exists \pi \in \text{Aut}(G) : |\pi| < |\bar{\pi}|$

Proof. The permutation $\bar{\pi}$ in cycle notation (including singletons), contains exactly all the edges of G , i.e.,

$$|G| = \sum_{c \in \text{cyc}(\bar{\pi})} |c|.$$

In G^π there is exactly one edge from each cycle of $\bar{\pi}$, hence $|G^\pi| = \sum_{c \in \text{cyc}(\bar{\pi})} 1$. From the definition of \mathcal{SC}

$$|\pi| < |G| - |G^\pi| = \sum_{c \in \text{cyc}(\bar{\pi})} |c| - \sum_{c \in \text{cyc}(\bar{\pi})} 1 = \sum_{c \in \text{cyc}(\bar{\pi})} |c| - 1 = |\bar{\pi}|.$$

\square

II. NEAR SYMMETRY-COMPRESSIBLE GRAPHS

Graphs arising in practice rarely exhibit (non-trivial) symmetries, making them non-compressible using the representation (π, G^π) . In this section we present an extension to the class of graphs \mathcal{SC} , which includes also many graphs without global symmetries, making it a more viable possibility for compression of realistic graphs.

A graph G might not exhibit any significant symmetries, but with a few modifications many new symmetries might arise, resulting in a symmetry-compressible graph H . Let us first give the representation of the allowed transformations of G . We restrict the modifications to adding and removing edges. Both adding and removing an edge can be described by simply specifying the edge; if the edge is present in the graph it is a deletion, otherwise an addition of the edge. The entire set of transformations from G to H can be described as the symmetric difference of both graphs, i.e., $G \oplus H$.

Of course any graph can be transformed into a symmetry-compressible graph (e.g. a clique), but we are interested in graphs where the transformation is worthwhile, in the sense that it can result in a smaller representation.

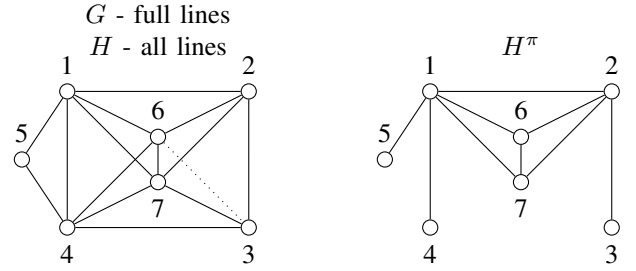


Fig. 1. A graph which has no symmetries, but is \mathcal{NSC} . The dotted line shows the edge that, when added, induces new symmetries in the graph, one of them, namely (14)(23), reduces the representation size by 2, since $|G| = 14$, but $|\pi| + |H^\pi| + |G \oplus H| = 2 + 9 + 1 = 12$.

Definition 3 (Near Symmetry-Compressible graph). A graph is near symmetry-compressible ($G \in \mathcal{NSC}$) if $\exists H$ and $\exists \pi \in \text{Aut}(H)$ such that

$$|\pi| + |H^\pi| + |G \oplus H| < |G|.$$

A simple example of a graph in \mathcal{NSC} is shown in Figure 1. The following theorem establishes a relation between the classes \mathcal{SC} and \mathcal{NSC} .

Theorem 4. $H \subseteq G, H \in \mathcal{SC} \implies G \in \mathcal{NSC}$.

Proof. We know that $\exists \pi : |\pi| + |H^\pi| < |H|$ (since $H \in \mathcal{SC}$). We can show that $G \in \mathcal{NSC}$ by noticing that $|G \oplus H| = |G| - |H|$, from which follows:

$$|\pi| + |H^\pi| + |G \oplus H| = |\pi| + |H^\pi| + (|G| - |H|) < |H| + (|G| - |H|) = |G|,$$

which is exactly the definition of near symmetry-compressibility. \square

III. ALGORITHMS

In this section we focus on practical compression algorithms, based on the notions of \mathcal{SC} and \mathcal{NSC} . The first algorithm searches for small compressible patterns inside the graph, whereas the second algorithm is more general, finding arbitrary large compressible patterns of a particular type.

TABLE II

SYMMETRY COMPRESSIBLE GRAPHLETS. WE GENERATED ALL CONNECTED GRAPHS OF SIZE 4-9, THE NUMBER OF SUCH GRAPHLETS IS GIVEN IN COLUMN 2. FOR EACH OF THESE GRAPHLETS WE CHECKED IF IT IS SYMMETRY-COMPRESSIBLE. THE NUMBER OF SUCH GRAPHLETS IS GIVEN IN COLUMN 3. COLUMN 4 GIVES THE AVERAGE RELATIVE EFFICIENCY AND COLUMN 5 GIVES THE MAXIMUM RELATIVE EFFICIENCY.

#	all	SC	avg. Δ^r	max. Δ^r
4	6	3	0.53	0.67
5	21	13	0.49	0.75
6	112	87	0.43	0.8
7	853	649	0.39	0.86
8	11117	7254	0.34	0.88
9	261080	126221	0.29	0.9

A. Graphlet based compression

In Theorem 4, we showed that graphs containing SC subgraphs are \mathcal{NSC} . This fact is used here to obtain a simple and practical algorithm. Namely, we focus on small graphs (a.k.a. graphlets). To do this we first find the symmetry-compressible subset of graphlets with 4 to 9 vertices. Table II summarizes the results, showing that a significant number of graphlets can be compressed. To give an idea of how much these graphlets can be compressed, we also computed the average and the maximum relative compression efficiency on these sets.

The algorithm uses a list of these graphlets as the basis for the compression. For each graphlet G' in this list, a maximal set of edge-disjoint subgraphs $H \subseteq G$ that are isomorphic to G' are found. All such subgraphs H are removed from G and their symmetry representation (π, H^π) is used instead. Algorithm 1 gives a more detailed description of the described procedure. To achieve better results (a higher compression ratio), a heuristic rule is used. Namely, the list of graphlets is initially sorted by decreasing relative efficiency. The intuition behind this heuristic is that we need to find the most compressible patterns first, otherwise the removal of subgraphs makes such patterns less likely to occur.

The presented algorithm uses the search for subgraphs extensively. Even though, in theory, this is a computationally intractable problem, many practical approaches exist [5], [6]. These algorithms work well also on large instances of graphs, especially when pattern graphs are small. And this is exactly the context we are using it in our algorithm, so the subgraph isomorphism is tractable for our application.

Algorithm 1 Compressing the graph with symmetric graphlets.

```

function GRAPHLETCOMPRESS( $G$ )
   $Comp = []$ 
  for all  $(\pi, G') \in Graphlets$  do
    while  $\exists H \subseteq G : H \cong G'$  do
      let  $\pi_H$  be the automorphism for  $H$  yielded by  $\pi$ 
       $Comp = Comp + (\pi_H, H^{\pi_H})$ 
       $G = G \setminus H$ 
  return  $Comp + G$ 

```

B. Compression with complete bipartite graphs

Here, we pick the family of complete bipartite graphs, which have simple symmetries that also have a large relative efficiency. The algorithm searches for such subgraphs repeatedly, using a similar approach as the graphlet compression.

We denote complete bipartite graphs on two vertex sets U, V as $K(U, V)$. Let $|U| = u$ and $|V| = v$. The size of the complete bipartite graph $|K(U, V)| = uv$. We will also use the notation $G(U, V)$ to signify the bipartite subgraph of G on vertex sets U and V . There are many symmetries in $K(U, V)$, we will use only one, which works well for any choice of u and v . The symmetry used henceforth is $\pi = (n_1 n_2 n_3 \dots n_u)$, where $n_i \in U$, therefore $|\pi| = u - 1$. The graph $K(U, V)^\pi$ are all the edges from one vertex of U to all vertices of V , i.e., $|K(U, V)^\pi| = v$. This results in the representation of size $v + (u - 1)$, which (for all $u, v \geq 2$) is smaller than uv , i.e., the size of the original graph.

Realistic graphs do not necessarily contain (large) complete bipartite subgraphs. However, they often contain large dense subgraphs, which can be transformed into $K(U, V)$ with only a few added edges. For such almost complete subgraphs, we first explore how dense must such subgraph be, in order to still be \mathcal{NSC} .

For an arbitrary bipartite graph G' (on two vertex sets U and V), we can check if it is potentially compressible (i.e., $\in \mathcal{NSC}$) as follows:

Theorem 5. A bipartite graph G' , on two vertex sets U, V where $|U| = u, |V| = v$, is \mathcal{NSC} if $|G'| > \frac{v+u-1+uv}{2}$.

Proof. The property follows from the definition of \mathcal{NSC} graphs, namely $|K(U, V)^\pi| + |\pi| + |G' \oplus K(U, V)| < |G'|$. And since $|K(U, V)^\pi| = v, |\pi| = u - 1$, and $|G' \oplus K(U, V)| = uv - |G'|$, the above inequality can easily be derived. \square

The general idea of the algorithm is to repeatedly find bipartite subgraphs having the above property, remove them, and represent them with $(K(U, V)^\pi, \pi, G(U, V) \oplus K(U, V))$. To find such a bipartite subgraph, the algorithm searches only for specific bipartite graphs, obtained in the following way. For each vertex $v \in G$, a bipartite graph is extracted. Its set U is simply the set of neighbors of v , whereas the set V is the set of the neighbors of vertices in U (which are not already in U). The extracted graph is then greedily optimized, to obtain a bipartite graph with a better relative efficiency, which for such graph is defined as

$$\Delta^r(G, U, V) = \frac{2|G(U, V)| - |V| - |U| + 1 - |U||V|}{|G(U, V)|}$$

The greedy optimization checks each vertex $v \in G(U, V)$ and computes Δ^r when v is present in the graph and when v (with all adjacent edges) is removed from the graph. If Δ^r is better in the later case, v is removed from $G(U, v)$. This process is repeated until no further improvement can be made. Among all extracted subgraphs (for all vertices of G), the one with the largest relative efficiency is selected and removed from G . This process is repeated until the

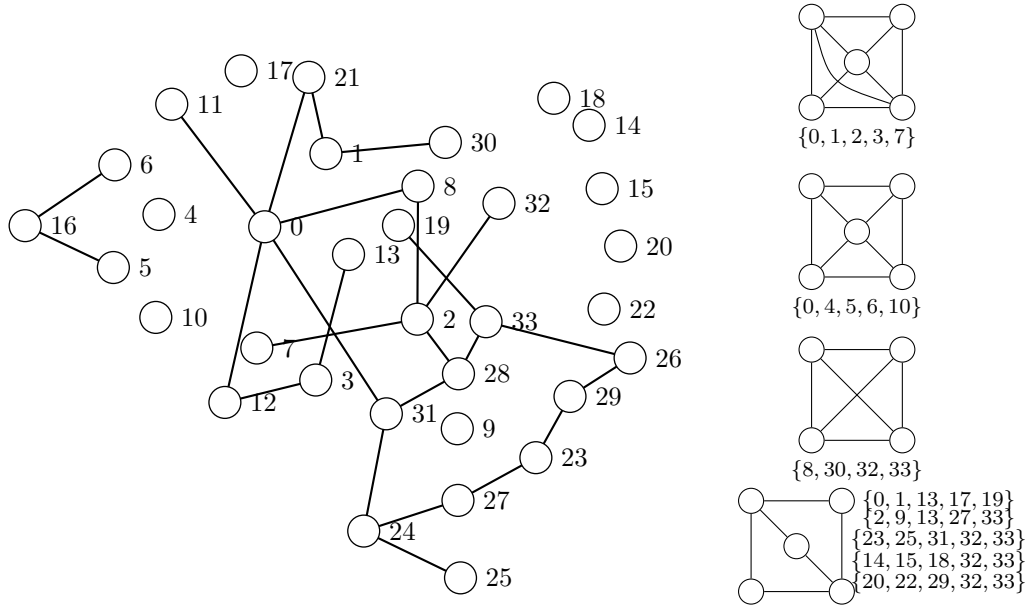


Fig. 2. The graph on the left is the remaining graph after the decomposition, using only the graphlets of sizes up to 5. The four graphlets that construct the final graph are shown on the right, with the corresponding vertex set. The last graphlet appears 5 times. The size of the residual graph is 25 edges, the size of the first graphlet and the symmetry is $3 + 2$, for the second graphlet is $2 + 3$, the third graphlet $2 + 2$, and the fourth graphlet $5(3 + 1)$. All together the size of the representation is 59 pairs.

best relative efficiency becomes non-positive. A more detailed description is given in Algorithm 2.

Algorithm 2 Compressing the graph with dense bipartite graphs.

```

function BIPARTITECOMPRESS( $G$ )
     $Comp = []$ 
     $B = \{\text{EXTRACTBIPART}(G, v) \mid v \in V(G)\}$ 
     $U, V = \arg \max_{(U, V) \in B} \Delta^r(G, U, V)$ 
    while  $\Delta^r(G, U, V) > 0$  do
         $H = G(U, V)$ 
         $Comp = Comp + (H, \pi_{U, V}, K(U, V) \oplus H)$ 
         $G = G \setminus H$ 
         $B = \{\text{EXTRACTBIPART}(G, v) \mid v \in V(G)\}$ 
         $U, V = \arg \max_{(U, V) \in B} \Delta^r(G, U, V)$ 
    return  $Comp + G$ 

function EXTRACTBIPART( $G, v$ )
     $U = N_G(v)$ 
     $V = \bigcup_{u \in U} N_G(u) \setminus U$ 
    while  $\Delta^r(G, U, V)$  is improving do
        for all  $v \in U$  and  $v \in V$  do
            if  $\Delta^r(G, U, V) < \Delta^r(G, U \setminus \{v\}, V)$  or  $<$ 
             $(\Delta^r(G, U, V \setminus \{v\}))$  then
                 $U = U \setminus \{v\}$  or  $V = V \setminus \{v\}$ 
    return  $(U, V)$ 
    
```

IV. EMPIRICAL EVALUATION

To demonstrate the practical potential of the presented concepts and algorithms, we devised an empirical evaluation on a set of real graphs. Most of these graphs are well known and used extensively in the various studies of network sciences. To demonstrate the potential also in image compression, we generated one example ourselves. The five well known graphs are Zachary's karate club graph [7] (karate), yeast protein

interaction network [8] (yeast), jazz musicians network [9] (jazz), US electrical power-grid network [10] (powergrid), and the facebook ego network [11] (facebook). We generated one graph from a binary image of Lena (lena), which is one of the most well-known images in computer science. All the graphs are undirected and were selected because they come from a variety of different real applications and are also structurally diverse.

For each of these graphs we ran the two described algorithms. Table III summarizes the obtained results. We compared the sizes of the uncompressed graph $|G|$, with the sizes of the obtained representations by the two algorithms (graphlets and bipartite). We can first see that the two algorithms have a comparable efficiency, but for practical applications, the bipartite compression is favourable, since its running time is significantly better. The obtained results divide the tests into two groups. The first three graphs (karate, yeast, powergrid) are less dense graphs, thus their reduced size is not that significant (31%, 25%, 8% respectively). The powergrid is very close to a tree graph, and we showed that trees are not \mathcal{SC} , so this result is not surprising. The second group of graphs (jazz, lena, facebook) shows a much better reduction in size (55%, 49%, 54% respectively). For an example of the compression, Fig. III-A shows the karate club graph compressed with graphlets.

V. CONCLUSIONS

Symmetries in graphs give a compact representation of self-similarities in graphs, which can be used to reduce redundant information in the classical representations of this type of data. In this article we formalized this intuition, introducing a new class of graphs, which we call symmetry-compressible graphs.

TABLE III

THE EMPIRICAL EVALUATION OF THE TWO PRESENTED ALGORITHMS. THE SECOND COLUMN GIVES THE SIZE OF THE GRAPH, THE THIRD AND FOURTH COLUMN THE SIZES OBTAINED WITH THE GRAPHLETS AND BIPARTITE ALGORITHMS RESPECTIVELY, AND THE LAST COLUMN GIVES THE BEST RELATIVE EFFICIENCY OBTAINED FOR THE GIVEN TESTSET.

dataset	$ G $	graphlets	bipartite	best Δ^r
karate	78	54	55	31 %
yeast	6650	4969	5372	25%
powergrid	6594	6046	6261	8%
jazz	2742	1246	1306	55%
lena	1339	678	705	49%
facebook	88234	44559	40457	54%



Fig. 3. Binary version of the famous Lena image.

Since graphs arising in real applications do not exhibit (m)any symmetries, we defined an extended class, near symmetry-compressible graphs, which becomes suitable for practical usage.

Since this is a new concept, many open questions remain. From a theoretical point of view, new properties of SC graphs can be discovered, and especially a deeper correlation between SC and $\mathcal{N}SC$ graphs must be established. Furthermore, the computational complexity of determining whether a graph is

in SC remains unanswered. Due to a deep correlation with the graph isomorphism problem, this problem is probably GI -hard, but the exact hardness must be determined by correlating it to some of the many similar problems in computational group theory.

From a practical point of view, the usability of the presented concepts must be explored more thoroughly. The global nature of the graph compression has the potential of improving the existing data compression algorithms, either as a preprocessing step, or as the central part of the compression. In our empirical evaluation we demonstrated how much a binary image can be compressed by using a graph representation, we believe that by combining this result with a more traditional image compression technique, better compression can be obtained.

REFERENCES

- [1] U. Čibej and J. Mihelič, "Symmetry-compressible graphs," in *2017 Data Compression Conference (DCC)*, April 2017, pp. 435–435.
- [2] G. Buehrer and K. Chellapilla, "A scalable pattern mining approach to web graph compression with communities," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008, pp. 95–106.
- [3] M. Adler and M. Mitzenmacher, "Towards compressing web graphs," in *Data Compression Conference, 2001. Proceedings. DCC 2001*. IEEE, 2001, pp. 203–212.
- [4] T. Suel and J. Yuan, "Compressing the graph structure of the web," in *Data Compression Conference, 2001. Proceedings. DCC 2001*. IEEE, 2001, pp. 213–222.
- [5] U. Čibej and J. Mihelič, "Improvements to Ullmann's algorithm for the subgraph isomorphism problem," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 07, 2015.
- [6] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub) graph isomorphism algorithm for matching large graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.
- [7] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, pp. 452–473, 1977.
- [8] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang *et al.*, "Topological structure analysis of the protein-protein interaction network in budding yeast," *Nucleic acids research*, vol. 31, no. 9, pp. 2443–2450, 2003.
- [9] P. M. Gleiser and L. Danon, "Community structure in jazz," *Advances in complex systems*, vol. 6, no. 04, pp. 565–573, 2003.
- [10] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [11] J. Leskovec and J. J. McAuley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547.

Java Virtual Machine Educational Tools

Tomaž Dobravec

Faculty of Computer and Information Science

University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenija

Email: tomaz.dobravec@fri.uni-lj.si

Abstract—Java virtual machine is becoming an industry standard for running platform-independent programs. It is available for a wide range of different platforms - from very simple embedded systems to large sophisticated computers. In spite of the fact that JVM supports efficient execution of complex programs with performances comparable to those achieved by non-virtualised programs, the JVM architecture is quite simple and as such it has very high educative value. The details about the JVM internal mechanisms, memory management procedures and bytecode execution processes can help software and hardware developers to understand the operation and the structure of the computer at the lowest level. For system developers, this knowledge can be used as a good example of a fast and efficient working environment, while the programmers can use it to write optimized programs that are in line with the JVM capabilities.

In this paper, we present some behind-the-scenes details about JVM. We list selected tools that can be used in the production as well as in the educational process to learn about and to take advantages of JVM as an efficient working environment. We present programs like javap, Bytecode Visualizer, and PyJVMGui, with some verbose test cases.

I. INTRODUCTION

Due to the thoughtful design based on previous experiences and focused goals of the development team, Java is a simple, object oriented, robust, platform independent, safe and high performance programming language. Its popularity reflects in several different-purpose programming language indices which place Java to the very top of the scoreboard [1], [2], [3]. Regarding these facts it is not surprising that Java has become a very popular programming language in both, academic and industrial environment. To take advantage of all the Java benefits (and thus to write efficient and reliable programs), it is very important to understand the whole life cycle of a program, which encompasses writing Java programs, translating from Java to bytecode and executing prepared bytecode in Java Virtual Machine (JVM) environment. In most courses that teach Java as a programming language only the first part of this cycle is presented to the students, while the details about the behind-the-scene behaviour of translated program inside the JVM are omitted. Since we think that these details are very important for a comprehensive understanding of the Java world, we teach them in some of our courses.

The rest of the paper is organised as follows. In Section II we describe some details on internal Java Virtual Machine behaviour, which are crucial in understanding the execution of bytecode programs. In Section III we present three tools that we use in the educational process: javap, Bytecode Visualizer, and PyJVMGUI. In the Conclusion section,

we provide some thoughts on using these tools in the classroom.

II. INSIDE THE JVM

JVM is a stack-based virtual machine, which means that all the instructions are performed with the usage of the build-in stack – before the execution of an instruction the stack is used to store operands, after the execution it holds the value of the result. Besides the *stack*, which is for obvious reasons its most important part, JVM consists of three major components: the heap, the method area, and the registers [4]. The *heap* is a part of the virtual machine’s memory where the space for object is allocated during the program execution. This part of the memory is controlled by the garbage collector who actively follows the references to allocated objects and occasionally clears-up all unused memory. The *method area* is a part of the memory, where the bytecode (compiled java or other code) is stored. During the execution of the program, JVM reads instructions from this area. A special thread-based counter, called the program counter, points to the (i.e. holds the address of the) location where the next instruction to be executed is stored. It is worth mentioning that although there are several objects of the same class active in a running program, the bytecode of their methods is written only once in the method area.

The JVM stack is a complex data structure used to store the information about the running program. To prevent misuse of this information and to ensure even distribution of memory usage, each thread gets its own independent stack. For each method that is executed within a thread, a frame is created on a top of the thread’s stack. This frame is divided into three parts: execution environment, local variables table, and operands stack, as depicted in Figure 1.

To manipulate a running thread of a program, JVM uses only four *registers*, namely the PC, frame, vars, and optop, which point to the current instruction to be executed, to the execution environment of the current frame, to the array of local variables, and to the top of the operand stack, respectively. On the stack frame of a method being executed, there is a storage for local copies of these registers. The values of registers are stored when a sub-method is called (see Listings 1) and restored on return (see Listings 2).

III. THE TOOLS

In the following we present different tools that are used to reflect, inspect, debug and/or manipulate java bytecode. All

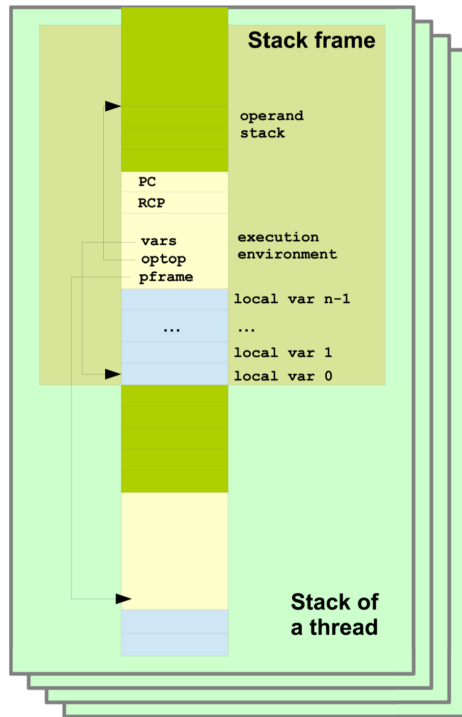


Fig. 1. The JVM stack of a thread

Listings 1. A call of method `b()` from method `a()`

- create `b_frame` on top of `a_frame`
- push method parameters (on `b_stack`)
- save `PC`, `optop`, `vars` to `a.PC`, `a.optop` and `a.vars`
- set `b.pframe = frame`
- `optop = b.optop`
- `vars = b.vars`
- `frame = b.frame`
- `PC = addr. of the first instr. of b()`

Listings 2. Return from method `b()`

- set `local_var_0 = result`
- `frame = b.pframe`
- `vars = frame.vars`
- `PC = frame.PC`
- `optop = frame.optop++`

the tools are publicly available and can freely be used in both industrial and educational environment. To demonstrate the features of the program we will use a simple java program (see Listings 3) that contains to static fields (`multiplier` and `divider`), method `sumEven()` and static method `main()`.

A. The `javap` program

For simple and quick look to the bytecode one can use the `javap` program, which is a part of standard JDK distribution.

Listings 3. Demo java program

```

public class Sum {
    static    int multiplier = 42;
    static final int divider  = 21;

    int sumEven(int[] tab) {
        int s = 0;
        for (int i = 0; i < tab.length; i++)
            if (tab[i] % 2 == 0)
                s += tab[i];
        return s;
    }

    public static void main(String[] args) {
        int[] numbers = {5,10,201,32769};
        Sum sum = new Sum();
        int c = sum.sumEven(numbers);
        c = c * multiplier / divider;

        System.out.println(c);
    }
}

```

To use a program, simply type `javap ClassName` or `javap -c -v ClassName`. The latest call gives a plethora of information about the class file, i.e., the version of `javac` that was used to compile the class (major and minor version), MD5 checksum, the name of the source file, constant pool values and program bytecode listings. Figure 2 shows a selected part of a constant pool of the `Sum` class. The usage of constant pool can be seen in the bytecode that is also listed by `javap`. For example, the bytecode that is generated from the line 16 of Listings 3 (`int c = sum.sumEven(numbers);`) contains four instructions, namely `aload_2`, `aload_1`, `invokevirtual 32` and `istore_3`. The first two instructions load the reference values of the second and the first local variables (i.e. `sum` and `numbers`) to the stack, the third instruction calls method 32 and the last instruction stores the result to the local variable number 3 (i.e. `c`). To decode the meaning of the constant 32 (i.e. to find out which method is called by the third instruction), we could use the constant pool presented in Fig. 2. The 32nd entry in the constant pool is a `MethodReference` that points to the 1st and the 33rd entry, which are the class name (i.e. `Sum`) and the `NameAndType` reference. The latter points again to the constant pool to the 22nd and 23rd entry, which contain the method's name and descriptor. Using the constant pool we found out that instruction `invokevirtual 32` will invoke the method `Sum.sumEven`, which receives an array of integers and returns an integer.

Another interesting information that can be derived from the bytecode is that JVM uses three different instructions to store the values in a newly created array, where the instruction

Constant Pool		
idx	tag	info
1	Class	name=2
2	Utf8	len=3; "Sum"
...		
5	Utf8	len=10; "multiplier"
6	Utf8	len=1; "I"
7	Utf8	len=7; "divider"
8	Utf8	len=13; "Constant"
9	Integer	bytes=21
...		
22	Utf8	len=7; "sumEven"
23	Utf8	len=5; bytes="([I)I"
...		
30	Integer	bytes=32769
...		
32	Method	class=1; nat=33
33	NAT	name=22; desc=23

Fig. 2. Selected parts of the constant pool for the Sum class

used depends on the value of the integer being stored in array. In particular, to store the values 10, 210 and 32769 to the numbers array, JVM uses instructions `bipush 10`, `sipush 210` and `ldc 30`, where 30 is an index to a constant pool entry, where the constant 32769 is stored. The difference between the first and the second instruction is that the first is encoded with two bytes, while the second is one byte longer.

The listed bytecode and constant pool reveals also the difference between final and non-final static variables. The multiplication with a non-final static variable is compiled into bytecode as `getstatic 13; imul` (i.e., load the value of an static field 13 (multiplier) to the stack and multiply with the other value on the stack), while the division with final static variable is compiled as `bipush 21; idiv`. It is obvious that the usage of final variables gives the compiler opportunity to create more optimised code.

B. Dr. Garbage's Bytecode Visualizer

Java Bytecode Visualizer (JBV) [5] is a tool used for visualizing and debugging Java bytecode. It is implemented as an Eclipse plugin which means that it can be used as additional view for the source files of the java Eclipse project. JBV offers two additional windows (see Figure 3) in which it displays bytecode listings accompanied by labels of source code lines and an outline of a flowchart of the entire program. Using all the information provided by JBV, user can debug a program and simultaneously compare source code with generated bytecode and observe the position of a program counter inside the flowchart.

```
int sumEven(int[] s) {
    /* L7 */
    0 iconst_0;
    1 istore_2;
    /* L8 */
    2 iconst_0;
    3 istore_3;
    4 goto 20;
    /* L9 */
    7 aload_1;
    8 iload_3;
    9 iaload;
    10 iconst_2;
    11 irem;
    12 ifne 9;
    /* L10 */
    15 iload_2;
    16 aload_1;
    17 iload_3;
    18 iaload;
    19 iadd;
    20 istore_2;
    /* L8 */
    21 iinc 3 1;
    24 iload_3;
    25 aload_1;
    26 arraylength;
    27 if_icmplt -20;
    /* L12 */
    30 iload_2;
    31 ireturn;
}
```

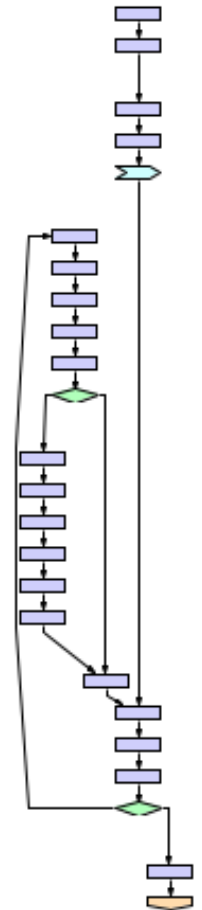


Fig. 3. Bytecode Visualizer view to the sumEven() method

C. PyJVMGUI

Another very useful program used to demonstrate the dynamics of execution of Java programs is called PyJVMGUI [6], [7]. The program which is based on PyJVM framework [8] was designed by Matevž Fabjančič at Faculty of Computer and Information Science, University of Ljubljana, as an educational tool to be used at the System Software course. The main intent of the program is to show the three main components of the JVM that are used during the execution of the program. Namely, the main window of PyJVMGUI shows the following information: java bytecode of the program (method) being executed, local variables table, operand stack and basic frame information. As an usage example of PyJVMGUI program let us observe the method `sumEven` of the `Sum` program. Figure 4 shows the content of PyJVMGUI window during the execution of this method. More precisely, in the left-side window, the bytecode of `sumEven` method is presented. The flow of the program is stopped in line 14 just after the instruction `irem` was fetched. This line of bytecode is a part

The screenshot shows the PyJVMGUI interface for Thread 2. The main window contains a table of instructions:

loc	Opcode	Operands
0	iconst_0	[]
1	istore_2	[]
2	iconst_0	[]
3	istore_3	[]
4	iload_3	[]
5	aload_1	[]
6	arraylength	[]
7	if_icmpge	[23]
10	aload_1	[]
11	iload_3	[]
12	iaload	[]
13	iconst_2	[]
14	irem	[]
15	ifne	[9]
18	iload_2	[]
19	aload_1	[]
20	iload_3	[]
21	iaload	[]
22	iadd	[]
23	istore_2	[]
24	iinc	[3, 1]
27	goto	[-23]
30	iload_2	[]
31	ireturn	[]

On the right side, there are several panels:

- Step**: A button to step through the code.
- Step Out**: A button to step out of the current method.
- Step until thread done/blocked**: A button to step until the thread finishes or is blocked.
- Frame**: Shows the current frame: InvVirt: sumEven ([I]) in Sum.
- Operand stack**: Shows the current stack contents:

Operands
32769
2
- Locals & Args**: Shows the local variable table:

Index	Value
0	(ref, 1442)
1	(ref, 1441)
2	10
3	3

Fig. 4. The main window of PyJVMGUI program during execution of the `sumEven` method

of the `if (tab[i] % 2 == 0)` Java clause in which we check if the array element is even.

The `irem` takes the top and the next-to-top elements from the operand stack and calculates the remainder of dividing the former with the later. In our case the stack contains elements 2 and 32769 (see the Operand stack window in Figure 4), which were pushed there by previous instructions `iaload` (which pushed an array element) and `iconst_2` (which pushed number 2). The `if` clause being observed is a part of a `for` loop, which is driven by the counter `i`. The method `sumEven` has one formal parameter (`tab`) and two local variables (`s` and `i`). Since `sumEven` is not a static method, the first element of the local variable table represents a reference to the current object (`this`) followed by the formal parameters and the local variables (see the Locals & Args window in Figure 4). This makes the variable `i` as the fourth element (i.e., the element with the index 3) in this table. Current value of `i` is 3, which means that the program in our example is stopped in the last iteration of the `for` loop. Note that this observation is also compliant with the fact that the stack contains the number 32769, which is the last element of the `tab` array. After the execution of the `irem` the stack will contain the result of the operation (in this case 1, since the remainder of dividing 32769 by 2 is 1), which will be used by the following instruction (`ifne`) in line 15, which performs a

9 bytes long jump (to line 27) if the first element on the stack is not zero. On the other hand, if the value is zero (i.e. if the division by 2 returned 0 as remainder, that is, if the number on stack is even), the instructions in line 18-24 will be executed (these instruction add the element value to the local variable with index 2). This short example shows how the information provided by PyJVMGUI can answer many question about the internal structure of the JVM. In our experience, the use of this program in the teaching process could greatly improve the understanding of the JVMs operation and virtual machines in general.

IV. CONCLUSIONS

In this paper we presented some details of JVM environment and selected tools that can be used to bring students closer to the Java behind-the-scene world. If we supplement these details with the knowledge of assembling and disassembling, we gain a comprehensive understanding of Java, which enables us to write quality programs.

For the last ten years, we have been teaching the backgrounds of the program's execution process as a part of the System Software course at the Faculty of Computer and Information Science in Ljubljana. Besides the extensive presentation of the static world (in which the programs to be executed are first compiled/assembled, then linked and finally loaded into the

memory) we also present a more recent dynamic approach in which the linking phase is done just before the usage of the entity concerned. A very good example of this concept is used in the Java Virtual Machine, where the linking is performed on-the-fly by using the location values written in the Runtime Constant Pool. This JVM mechanism enables the usage of the programs in which not all the functionality is already present and it might be added later. To make the learning process more interesting and instructive, we have been using some dedicated tools [9], [10] for several years. These tools were developed especially for the needs of the System Software course and they help to present the behaviour of static world programs in the educational process. Our experiences show that the usage of these tools is a great encouragement for students that not only makes the learning process faster and more interesting but it also inspires and motivates students to dig into the lowest levels of computer architecture. Based on good experiences with the static world behaviour we wanted to provide a similar environment in the dynamic JVM world. Unfortunately, we found out that there are very few tools that could be used to present the dynamic execution of the bytecode in the virtual machine. The programs, that enable most of the desired functionality, are the `javap` and `Bytecode Visualizer`. They both present the bytecode of a given program and some accompanying parameters (like constant pool, method table and the like). Even though the latest also enables dynamic monitoring of a program being executed (in debug mode), it does not show the internal JVM structure with the linking mechanisms. To bridge this gap, we have developed our own program (`PyJVMGUI`) that shows the behaviour of the JVM while executing the bytecode of a given program (e.g. the usage of methods stack, operands stack, local variables table, and JVM internal registers). Since this program was developed recently, we have not used it in the classroom yet, so we can not report the results on its usage. But according to our experiences with the `SicVM` tool and based on the opinion of selected students that have already tested and evaluated the program, we can reasonably expect that the usage of this program in the educational process will encourage students to deepen into a very interesting world of the JVM and thus to gain and to expand the knowledge of the computer logic at the lowest level.

REFERENCES

- [1] T. S. BV, “Tiobe index (june 2019),” www.tiobe.com/tiobe-index/, 2019.
- [2] P. Carbone, “Pypl popularity of programming language (june 2019),” pypl.github.io/PYPL.html, 2019.
- [3] S. O’Grady, “The redmonk programming language rankings: January 2019,” redmonk.com/sogrady/2019/03/20/language-rankings-1-19/, 2019.
- [4] B. Venners, “The lean, mean, virtual machine,” *JavaWorld*, 1996.
- [5] D. G. Community, “Dr. garbage tools,” <http://drgarbagetools.sourceforge.net/>, 2019.
- [6] M. Fabjančič, “Simulating Java bytecode execution (in Slovene language),” diploma thesis (mentor: T. Dobravec), Faculty of Computer and Information Science, University of Ljubljana, 2018.
- [7] —, “A GUI for the PyJVM,” <https://pypi.org/project/pyjvmgui/>, 2018.
- [8] A. Romanenco, “PyJVM - Java 7 virtual machine implemented in pure python,” <https://github.com/andrewromanenco/pyjvm>, 2014.
- [9] K. KLOBOVES, J. MIHELIČ, P. BULIĆ, and T. DOBRAVEC, “FPGA-based SIC/XE processor and supporting toolchain,” *International journal of engineering education*, vol. A, no. 6, pp. 1927–1939, 2017.
- [10] J. MIHELIČ and T. DOBRAVEC, “Sicsim : a simulator of the educational sic/xe computer for a system-software course,” *Computer applications in engineering education*, vol. 23, no. 1, pp. 137–146, 2015.

An implementation of the *M-tree* index structure for PostgreSQL using GiST

István Donkó*, János M. Szalai-Gindl†, Gergő Gombos‡ and Attila Kiss§

Department of Information Systems

Eötvös Loránd University

Budapest, Hungary

Email: { *isti115, †szalaigindl, ‡ggombos, §kiss }@inf.elte.hu

Abstract—Several index structures are competing for efficient operations on various types of data, but there are some forms of information that cannot fit into the existing models, because by forcing them into one of the currently available representations they either lose some of their value or they do not have the necessary properties which would allow them to be used in the first place. In light of this, we implemented the *M-tree* [1] [2] index structure under the PostgreSQL database management system through a GiST [3] extension to facilitate efficient range-based queries and *k*-nearest neighbor searches on data that resides only in a metric space. The *M-tree* structure has parts that require heuristics, of which the implementations are not entirely specified, only suggestions are given. This allows flexibility for adapting to different situations and leaves space for further improvements. We implemented several of the strategies proposed in the original paper alongside some of our own ideas and saw significant speed gains when compared to a sequential scan, our main reference, as the other built-in index structures are not applicable to metric spaces.

Index Terms—Databases, Indexes, *M-tree*, Metric space, Index structure, PostgreSQL, GiST, Range search, *k*-NN search

I. INTRODUCTION

The initial motivation was being able to efficiently search for similar entries in a database relying only on a measurement of difference. For example, when gathering musical information that is periodically sampled for notes, there can often be embellishments that are picked up in a recording and not in another one, thus offsetting the entire entry and making direct comparison ineffective. The advantage of edit distance (also known as Levenshtein distance) is that it does not depend on data being exactly aligned, by defining the difference between two objects in terms of the number of insertions, deletions and substitutions needed to transform one into the other. It also allows the comparison of data entries with different lengths. To achieve this we needed an index structure that does not depend on the existence of an ordering on the entries or the ability to represent the given information in a vector space. The idea of the *M-tree*, as introduced by [1] and [2] seemed like a perfect fit for this purpose as it only requires the distance function to adhere to the properties of metric spaces, which are satisfied by the chosen calculation method.

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). This research was supported by Thematic Excellence Programme, Industry and Digitization Subprogramme, NRDI Office, 2019. Supported by the ÚNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology.

The C and SQL source codes of the project are available at: <https://www.github.com/isti115/postgresql-gist-mtree>.

While there are other existing types of tree structures that are capable of handling metric data with different approaches, we could not use them as reference points when evaluating our result, as they do not have readily available implementations under PostgreSQL. A few examples for such, which could be implemented in the future and compared against:

- BK-tree [4] [5], a solution that specializes in distance metrics with integer values in a preferably small range (often used with edit distance), so that the nodes can be distributed into groups based on their distance to their ancestor.
- Slim-tree [6], which is explicitly designed based on the idea of avoiding overlaps caused by nodes with a high “*fat-factor*” (hence the name) using efficient insertions and node splits to improve query performance.
- VP-tree [7] and its variations, multiple VP-trees [8] and VP forests [9], that are built on the concept of partitioning the metric space in terms of distance thresholds from specific entries selected as so-called “*vantage-points*”. The separation results in two groups for each vantage-point, one that is closer than the threshold and the other one that is further away. When compared to the *M-tree*, they seem to produce smaller overlaps, but turn out to be less scalable and dynamic. [10]

The rest of the paper is structured in the following way: Section II describes the concepts and technologies utilized in the implementation, then Section III discusses the achieved results, after which Section IV presents some use cases where the developments would be applicable and finally Section V draws the conclusions and poses some options for further improvement.

II. TECHNOLOGIES USED

A. *M-tree*

The main index structure of the *M-tree* is described in [1]. It is based on the idea of attaching a so-called covering radius to the nodes in the search tree that represents the maximal distance of any of the entries in any subtree of that branch. A simple illustration can be seen in Fig. 1, which displays the tree structure using arrows, shows the covering radii belonging to

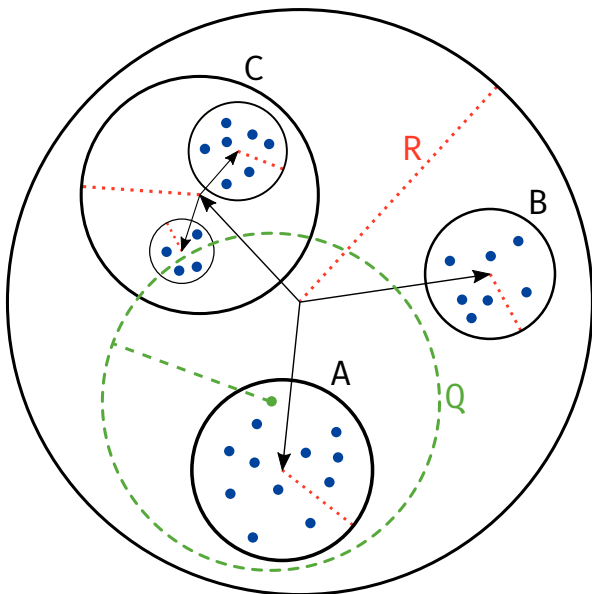


Fig. 1. Illustration of the M-tree structure, where blue dots are representing the entries storing the actual information, the red dotted lines (the longest of which is denoted by "R") are the covering radii of the nodes and the green dashed region (denoted by "Q") is an example for a range-based query.

the nodes with red dotted lines (the radius of the largest node is marked by "R") and has some arbitrarily placed blue dots to represent the stored information. Do not be confused by the fact that this image resides in a two-dimensional space, there is no absolute position for the nodes, neither for the entries, only their relative distances can be calculated.

By indexing data in this manner, the opportunities to speed up queries arise by being able to make certain decisions during lookup without visiting all the leaves. For example if an entire subtree (marked by the letter "A" in Fig. 1) fits inside the query range that is currently being searched (marked by "Q" in the figure), because its node is closer to the center of the query than the radius of the query minus the radius of the node, all of its content can surely be included in the results. On the other hand if it can be decided that an entire subtree (marked by the letter "B" in the figure) is unable to satisfy a condition, because its node is further apart from the center of the query than the sum of their radii, thus the chance of an overlap can be ruled out, so none of its elements can match the query, that branch can be completely cut off from the search. In the third case, where a subtree (marked by letter "C" in the figure) is neither completely inside, nor fully outside the search region, its children have to be processed further.

B. PostgreSQL

We chose PostgreSQL¹ as the database management system to extend because of its support for GiST (Generalized Search Tree) based index development² that fit the needs and constraints of the M-tree architecture. Using the `varlena`

¹<https://www.postgresql.org/>

²<https://www.postgresql.org/docs/current/gist-extensibility.html>

extensible struct (a variable-length array type that stores its size in the first few bytes and its contents in the rest of its allocated space) as a base, we defined our own data type that includes a covering radius and the actual data in the flexible array member.

C. GiST

The GiST data structure is an abstract tree structure that can be considered as a generalization of other trees, such as the more well known B-tree and R-tree or in this case the M-tree. It enabled the extension of the system with a new index type. For that, we needed to implement several functions.

The ones that belong to the index creation process were:

- `union`: Given a set of entries it creates a single node that represents all of them by selecting one of them and assigning a covering radius to it that is equal to the distance of the furthest object from the selected one and thus contains all the entries. We chose to implement it in a way that tries to select the node closest to the imaginary *center* (only intuitively, as such a center does not exist in a metric space), meaning it has the lowest possible maximal distance to every other node, so that the covering radius will be minimal.
- `picksplit`: When the number of a node's descendants reaches a certain threshold set by the database management system (usually in connection with the page size), it needs to split that node and distribute its children between the newly promoted items. This algorithm is responsible for selecting the two nodes that should be promoted in a way that ideally would later benefit the queries, for example by minimizing the overlap between the two covered areas.
- `penalty`: This method calculates how much the efficiency of the index would drop if an item was to be inserted under a certain node. This is estimated by calculating the required increase in covering radius to keep the correctness of the index. When inserting a new item the values provided by this function contribute the information needed for choosing a good path.

The other methods being utilized during queries:

- `consistent`: The purpose of this method is to determine whether a key that is stored in a node of the tree is *consistent* with the current query, meaning that its subtrees can potentially contain leaves that match the specified criteria. If this test is negative, then the whole branch can be pruned from the actual search tree. There is actually also a second value (named `recheck`) returned that serves the purpose of avoiding unnecessary comparisons by allowing the implementation to tell the system when it can be determined from the observation of only one node that all the leaves belonging to its descendants must satisfy the given condition, thus do not need to be tested individually and that entire branch can be included in the results.
- `distance`: As M-trees operate on values residing in metric spaces there always exists a distance function that



Fig. 2. Example sheet music for the beginning of the popular English lullaby, "Twinkle, Twinkle, Little Star".

can determine how far two values are from one another. By providing this function we allow the index structure to be used for ordering operators, which is essential for k -nearest neighbor queries as that needs to find entries based on their distance to a given *center* value, thus compare their separation from the midpoint to each other.

There would have also been four optional functions that could have been implemented (and can potentially be later added if needed) that were not necessary for our use case:

- *same*: If the representation of the data would not be injective with regards to its actual contents then this function could serve the purpose of defining when two entries should be considered equal.
- *compress*: In cases where it would be desirable to lower the amount of storage being used by the index structure, the internal tree nodes could use a smaller type than the actual data by implementing a compression algorithm in this method.
- *decompress*: When compression is applied to the stored data this function would provide the means of reconstructing the information that is needed by the other methods. (That is not necessarily the full original data in case of internal nodes.)
- *fetch*: In contrast to *decompress*, this function would be responsible for fetching the entire original information in a lossless manner where it is required.

III. RESULTS ACHIEVED

We obtained test data in the form of ABC music notation³ (which seemed fitting for the task because of its simplicity) from a compilation⁴ made by professor José Fernando Oliveira⁵ from the University of Porto. This data was stripped down into a heavily simplified representation by only keeping the notes and disregarding all rhythm related information. Also, all the melodies were normalized to the key of C based on the included metadata, so for example the piece of sheet music that can be seen in Fig. 2 would become "ccggaag".

The set contained 19296 entries which we shuffled to avoid organizational grouping bias and then generated different sized test sets from. (To be exact, with 1000, 3333, 6666, 10000, 13333, 16666 and one with all the 19296 entries.) We then built the index on these datasets using different heuristics for the picksplit algorithm and measured their performance on range-based and k -nearest neighbor queries placed around randomly selected entries. After repeating this randomization

and measurement process several times we averaged the results and compared the different strategies to each other and the sequential scan as a baseline.

These benchmarks were executed on a machine equipped with an Intel® Core™ i5-650 CPU and 4GB of RAM running Ubuntu 16.04.

As already explained above, the goal of a picksplit algorithm is to choose two representative elements from the group of nodes to be split, among which when the rest are distributed, queries could be performed efficiently. The eight competing versions, including three deterministic, one completely random and finally four sampling type of strategies were:

- *PicksplitFirstTwo* – Simply choosing the first two entries in the list to be promoted.
- *PicksplitMaxDistanceFromFirst* – Searching through the list of entries and choosing the first one and the one that is furthest apart from it in hopes of creating as little overlap as possible.
- *PicksplitMaxDistancePair* – Iterating on the previous version, this searches for the two nodes in the list that have the highest possible distance out of all the combinations.
- *PicksplitRandom* – Choosing two entries randomly from the list.
- *PicksplitSamplingMinCoveringMax* – This strategy randomly picks two nodes, calculates the split that would occur if those two were to be promoted and stores the bigger covering radius of the two. After that, it repeats the procedure several (in our implementation a 100) times by taking another two nodes and overwriting the results if those produce a smaller maximal radius.
- *PicksplitSamplingMinCoveringSum* – Takes samples similarly to the previously described method, but instead of trying to minimize the bigger radius of the two, it aims to find the lowest possible sum of the radii.
- *PicksplitSamplingMinOverlapArea* – This strategy calculates the imaginary overlap between the two potential regions based on their radii and the distance between them, and aims to find a pair that produces a minimal intersecting area.
- *PicksplitSamplingMinAreaSum* – As a variation of the *PicksplitSamplingMinCoveringSum* strategy, instead of simply adding together the radii, this method calculates some sort of a collective *area* (as if the nodes were circles in two dimensions) defined by the sum of their squared radii.

As an example, Fig. 3 illustrates a situation in which two strategies would prefer different options. If presented with the possibility of splitting node A in the way it can be seen under B or C, the *PicksplitSamplingMinCoveringSum* strategy would favor B, as adding the two resulting radii together would result in a lower value there, while *PicksplitSamplingMinCoveringMax* would rather choose C, because the bigger radius of the two is smaller in that case.

³<http://abcnotation.com/>

⁴http://www.atrilcoral.com/Partituras_ABC/index_abc.htm

⁵https://sigarra.up.pt/feup/en/func_geral.formview?P_CODIGO=209980

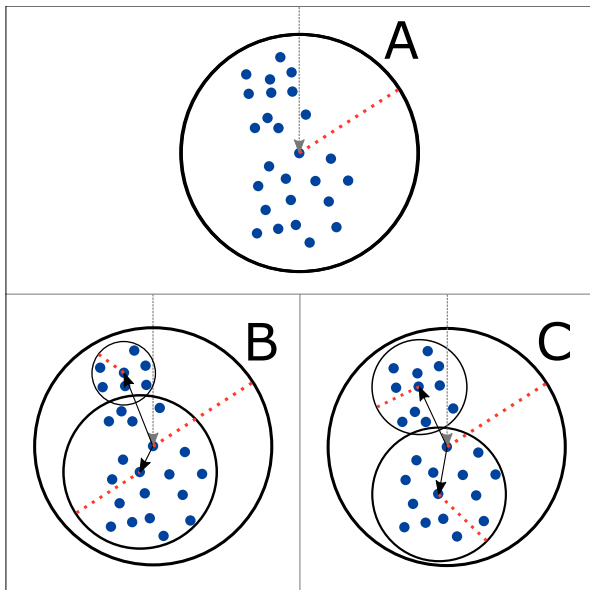


Fig. 3. Illustration of two possible options when distributing the contents of a node during a picksplit operation that are favored in two different heuristics.

When comparing these different possible heuristics for splitting we found that there was a visible trade-off between the time it takes to build an index and the query performance afterwards. Increasing the complexity of the algorithm responsible for selecting the nodes to promote generally resulted in a structure that was better suited for range-based queries. There was no significant difference when comparing the k -nearest neighbor performance, all the strategies far outperformed the sequential scan and were relatively close to each other.

As expected, the sequential scan performed linearly in terms of the amount of data in both tests, as it always had to traverse the entire database in order to provide the queried results.

A. Index building times

One of the drawbacks of our approach are the longer initial index building times, which are obviously nonexistent when doing sequential scans, but this is eased by the fact that one of the key advantages of the M-tree that [1] and [2] mention is the ability to handle dynamic data as opposed to other index structures which sometimes can only operate efficiently on static information. Compared to other spatial index structures M-tree seems more suitable for data that is expected to change, while some other approaches can only work on inert data (thus requiring a rebuild of the index whenever the underlying information changes), but in contrast to those the M-tree can also perform efficient insertions afterwards.

Usually the heuristic that minimizes the sum of the two covering radii (`PicksplitSamplingMinCoveringSum`) when splitting took the most time to build. Taking a look into its index creation process we found that it tends to produce deeper trees, because it favors uneven splits between the two promoted nodes, as one of the covering radii being small and

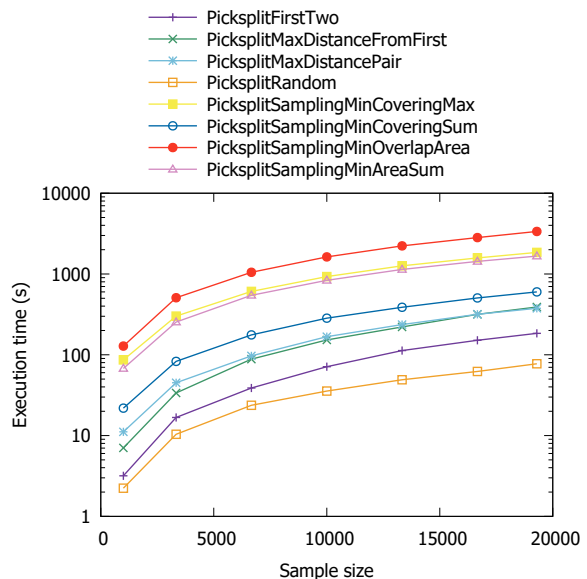


Fig. 4. The time it took to build the index with different splitting strategies for several data sample sizes.

the other one bigger seems to result in an overall lower sum than when the two are closer to being equal.

Most likely a similar reasoning explains what happens with `PicksplitSamplingMinOverlapArea`, since it can achieve zero overlap by setting one of the radii to zero, and this results in a heavily leaning tree.

The `PicksplitRandom` strategy seems to have the shortest index building time followed by the deterministic algorithms, while the sampling type of heuristics are slower, as those have to repeat the calculations over and over again.

B. Range-based queries

The execution times for range-based queries can be observed in Fig. 6. This first type of query that the M-tree supports is one where we would like to access the neighbors of a given entry within a certain radius. The execution times were greatly reduced for smaller radii, as there were fewer overlaps and greater sections of the tree could be avoided while searching. As expected, the performance benefits start to lessen when the radius is increased, as fewer branches can be pruned.

As it was already mentioned in the previous section, the `PicksplitRandom` strategy took the least time to build the index, which seems to relate to the quality of the structure. As can be seen, it was the one that provided the smallest amount of improvement during the range-based queries.

C. K -Nearest neighbor queries

As you can see in Fig. 7, in the k -nearest neighbor search the M-tree based index significantly outperformed the sequential scan. When we first saw the results, something that seemed strange was the fact that some picksplit strategies have seemingly managed to reduce their execution time when testing on a bigger data set. Our initial explanation for this was that when

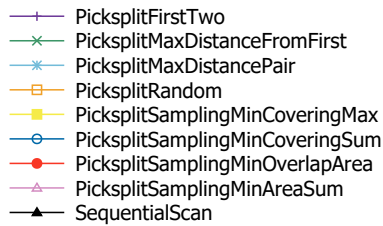


Fig. 5. Shared key for Fig. 6 and Fig. 7.

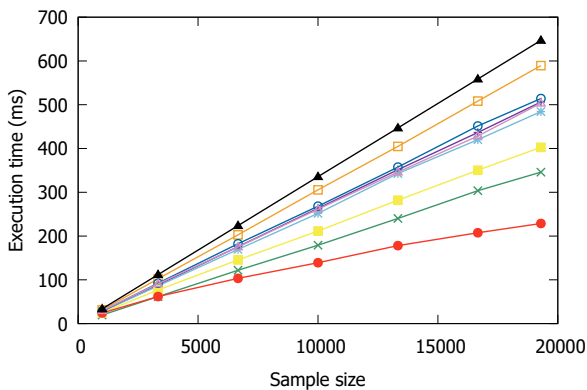
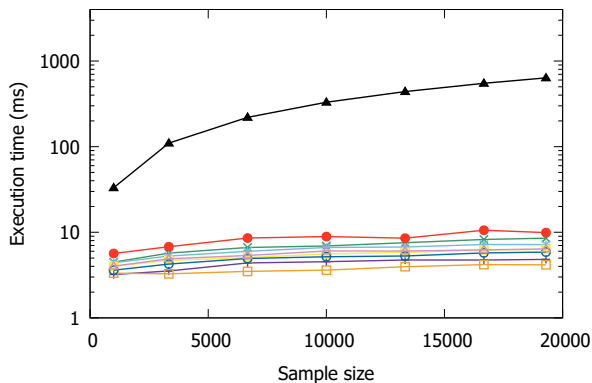


Fig. 6. The time it took to execute range-based queries using indexes built with different splitting strategies for several data sample sizes. We tested smaller and larger search limits and observed that the improvements are greater in case of a shorter radius, such as the one presented here. See Fig. 5 for key.

Fig. 7. The time it took to execute k -nearest neighbor queries using indexes built with different splitting strategies for several data sample sizes. See Fig. 5 for key.

there are more entries, the space is more densely populated, thus the neighbors are closer together, so fewer branches have to be traversed in order to find the same amount of neighbors, but after running multiple measurements it became clear that the fluctuations in the different data sizes are more likely due to noise, as measuring such low values (around 10ms) turns out not to be accurate enough. When averaging out the results of multiple measurements the plots started to flatten into a very slowly rising line.

We also measured query execution times for exact matches, because that was the only aspect in which we could possibly compare with other (non-metric-only) index structures, as

without defining an ordering (avoiding which is exactly the point of the M-tree) those would essentially perform equal to a sequential scan. When looking for an exact match, the M-tree was already outperformed by the sequential scan and even more so by a B-tree, but that is not surprising, as it was not designed for that use case. What we would suggest as the probable cause for that are the overlaps between the nodes with their covering radii, which forces the search to traverse multiple paths, as in other index structures overlaps (if they exist at all) are more easily avoidable. With that said, it does not mean that creating an index structure with our process will reduce the performance of any queries, as the PostgreSQL query planner chooses the sequential scan over our index unless explicitly being instructed to use the index instead.

IV. POSSIBLE USE CASES

In this section we present some possible real-life scenarios in which the system could prove to be useful by explaining what difficulties can occur and how this construction could contribute a solution to them.

A. Music

1) *Audio based music search*: Websites and applications, such as *SoundHound* and *Shazam* provide services that are capable of identifying a song from either a short recording of the actual audio or at times even by just having the user sing or hum part of the melody. As far as we know, these are currently based on fingerprinting and other similar techniques. It often occurs that somebody does not remember a whole sequence correctly and thus may leave out notes or maybe even add extra notes out of imagination. In these cases with an appropriate distance definition the efficient K Nearest Neighbor queries made possible by this implementation could help in the identification of the song by matching it against a database that has an M-tree index.

2) *Common origin research*: Staying within the context of music, range-based queries on the other hand could be utilized for the discovery of connections, for example between folk music of different nations that are based on the same melodies but are extended with different ornaments, by searching within a certain similarity range.

B. Genetics

In the field of genetics matching sequences to each other is a very important procedure, as it allows the construction and examination of theories that can give way to many conclusions to be drawn. By developing a distance metric based on the probabilities of certain types of mutations [11]^{6 7} extracted from existing statistics and biological understanding of these processes, an efficient lookup method could be created by building indexes on top of existing databases. A close example can be found in the article introducing ND-GiST [12], which is also an index structure created using GiST for the purpose of improving query performance on genetic data.

⁶<http://www2.csudh.edu/nsturm/CHEMXL153/DNAMutationRepair.htm>

⁷<https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations>

C. Publishing

All forms of media, where originality is a requirement often face the challenge of checking for already existing similar works. Exact copies are easy to filter, but when an adversary alters the given information with the intent of making it harder to detect more robust methods are necessary. For example in the case of plagiarism detection, where some material's authenticity, be it text, music, picture or other work needs to be asserted in relation to an existing database of entries, and an appropriate distance metric can be designed to discover similarities, an M-tree based construction is expected to perform quickly enough while resulting in a more tunable experience.

D. Spell checking and autocompletion suggestions

When typing plain text or source code it can be very convenient when by only typing a few letters a correct guess is presented, thus we do not need to type the entire word, just accept what was suggested. Several programs only offer words that contain an exact match to what we typed. Some other applications allow different parts of the desired string to be entered (e.g. only the end) to avoid having to type long parts that are the same between different suggestions, thus being unhelpful in deciding between them. What would be even better, but is rarely available in current software, is the so-called *fuzzy string search*, which is essentially based on a penalty system calculated using edit distance, thus is able to tolerate typing errors that add letters that are not present in the desired string. When the collection of words to choose from is sufficiently large enough (for example the words of an entire language), such a process could potentially be sped up by making use of a database with an M-tree index and k -nearest neighbor queries for suggesting the k closest alternatives.

V. CONCLUSIONS

The achieved results are already capable of providing significant performance improvements, but the field is far from extensively discovered and there are still many ways in which query times could potentially be further reduced, for example:

- Better tree splitting heuristics could be implemented that further optimize the resulting structure and by that improve the efficiency of the queries.
- In case of the *sampling* type of strategies measurements could be conducted to determine the tradeoff curve between smaller and larger sample sizes, as testing more potential pairs of promoted nodes certainly increases the index building times, but probabilistically should provide a structure that is more adherent to the rule of said strategy.
- Variations on certain existing splitting strategies could also be created by adjusting some of their parameters, such as trying different powers instead of `PicksplitSamplingMinAreaSum`, for example cubing the radii would essentially make it into `PicksplitSamplingMinVolumeSum` and so on into even higher dimensions.
- Multiple different strategies could be merged by weighting their scores. This way desirable properties could be combined (for example if one strategy incentivizes a favorable property while another penalizes something that is preferably avoided, these could be achieved together), resulting for example in a strategy that still aims to minimize overlap, but avoids zero-area nodes because of being persuaded to create even splits by giving it a penalty based on, say, the squared difference of the radii.
- The concept of a Key Dimension could be added from the M^+ -tree, which benefits highly dimensional data. [13]
- In cases where the calculation of the chosen metric is resource-intensive a caching mechanism could be added to reduce the number of distance computations.

Also, the current results could be compared to other ways of solving the same problem, for example the duplication search method for ABC notated data found in [14].

REFERENCES

- [1] P. Ciaccia, M. Patella, and P. Zezula, "M-tree an efficient access method for similarity search in metric spaces," in *Proceedings of the 23rd VLDB Conference Athens, Greece, 1997*. IBM Almaden Research Center: Very Large Databases Endowment Inc., 1997.
- [2] P. Ciaccia, M. Patella, F. Rabitti, and P. Zezula, "Indexing metric spaces with m-tree," in *SEBD*, vol. 97, 1997, pp. 67–86.
- [3] J. M. Hellerstein, J. F. Naughton, and A. Pfeffer, "Generalized search trees for database systems," in *Proceedings of the 21st International Conference on Very Large Data Bases*, September 1995, pp. 562–573.
- [4] W. A. Burkhard and R. M. Keller, "Some approaches to best-match file searching," *Commun. ACM*, vol. 16, no. 4, pp. 230–236, Apr. 1973. [Online]. Available: <http://doi.acm.org/10.1145/362003.362025>
- [5] R. Baeza-Yates and G. Navarro, "Fast approximate string matching in a dictionary," in *Proceedings. String Processing and Information Retrieval: A South American Symposium (Cat. No. 98EX207)*. IEEE, 1998, pp. 14–22.
- [6] C. Traina, A. Traina, B. Seeger, and C. Faloutsos, "Slim-trees: High performance metric trees minimizing overlap between nodes," in *Advances in Database Technology — EDBT 2000*, C. Zaniolo, P. C. Lockemann, M. H. Scholl, and T. Grust, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 51–65.
- [7] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '93. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1993, pp. 311–321. [Online]. Available: <http://dl.acm.org/citation.cfm?id=313559.313789>
- [8] T. Bozkaya and M. Ozsoyoglu, "Distance-based indexing for high-dimensional metric spaces," *SIGMOD Rec.*, vol. 26, no. 2, pp. 357–368, Jun. 1997. [Online]. Available: <http://doi.acm.org/10.1145/253262.253345>
- [9] P. N. Yianilos, "Excluded middle vantage point forests for nearest neighbor search," in *In DIMACS Implementation Challenge, ALENEX'99*, 1999.
- [10] M. Keyvanpour and N. Izadpanah, "Analytical classification of multimedia index structures by using a partitioning method-based framework," *The International Journal of Multimedia & Its Applications*, vol. 3, no. 1, 2011.
- [11] C. Tomasetti, "On the probability of random genetic mutations for various types of tumor growth," *Bulletin of mathematical biology*, vol. 74, no. 6, pp. 1379–1395, 2012.
- [12] J. M. Szalai-Gindl, A. Kiss, G. Halász, L. Dobos, and I. Csabai, "Nd-gist: A novel method for disk-resident k-mer indexing," in *World Conference on Information Systems and Technologies*. Springer, 2019, pp. 663–672.
- [13] X. Zhou, G. Wang, J. X. Yu, and G. Yu, "M+-tree : A new dynamical multidimensional index for metric spaces," in *ADC*, 2003.
- [14] C. Walshaw, "A statistical analysis of the abc music notation corpus: Exploring duplication," 06 2014.

Optimization design for parallel coloring of a set of graphs in the High-Performance Computing

Adam Dudáš
*Faculty of management science and
informatics*
University of Žilina
Žilina, Slovakia
adam.dudas@fri.uniza.sk

Jarmila Škrinárová
Faculty of Natural sciences
Matej Bel University
Banská Bystrica, Slovakia
jarmila.skrinarova@umb.sk

Eduard Vesel
*Faculty of management science and
informatics*
University of Žilina
Žilina, Slovakia
eduard.vesel@fri.uniza.sk

Abstract—*This paper presents solution to problem of edge coloring of sizable set of cubic graphs and examination of relations between these graphs. We solved this problem on various computing systems and for various sizes of the problem (various number of graphs). For the computations we used High-Performance Computing Cluster and Amazon Web Services cloud environment. We measured and analyzed time of computation of edge coloring and other properties. Largest set we worked with contained almost 10 million graphs. We created new methodology, which can be used to finding order of the edges which optimizes time of computation of edge coloring for certain subset of graphs. On the basis of this methodology, we implemented algorithm for parallel edge coloring of set of graphs. For testing of the methodology, we designed 8 experiments. Results showed, that worst time of edge coloring of graph from set of 19 935 graphs before use of the methodology was 1260 ms. After application of our methodology, we found same order of edge coloring for whole group of 19 935 graphs and the highest time of coloring was 10 ms.*

Keywords—parallel and distributed computing, HPC, virtual machines, edge coloring of cubic graphs

I. INTRODUCTION

This paper is motivated by the need for optimization of time of edge coloring of graph sets. Edge coloring is NP-complete problem [1] which invites use of high performance computing systems and specialized algorithms.

We developed new parallel algorithm for coloring sizable sets of graphs, which calls edge backtracking algorithm multiple times. The edge backtracking algorithm contains recursive functions for edge coloring of graphs. This coloring algorithm cannot be divided (whole computation of the algorithm needs to be performed sequentially). In other words, our parallel algorithm contains sequential parts which cannot be interrupted.

High performance computing systems currently work on the basis of physical machines (computing elements) or use virtualization through virtual machines. Use of virtual machines in the computing systems can cause latency, which can be reflected in significant slowdown of various computational tasks. [17]

Our first objective is to explore how does use of computational system with virtual machines affect slowdown of parallel algorithm which contains recursive functions.

Our second objective is to optimize (minimize) computation time of coloring of sizable sets of graphs. In our novel approach we search for order of edge coloring which optimizes the computing time of coloring of one graph and at the same time large set of graphs. We are looking for this order of edges with the use of random permutations of graphs, as shown in the algorithm presented in the Section IV.

We use set of graphs from House of graphs database [13] which consists of 19 935 34-vertex Snarks with cyclic 5-edge connectivity.

We designed and implemented methodology and created 8 experiments for time optimization of problem of edge coloring of graphs from selected set. Our work towards objectives of this paper is presented in the following order:

Second section of this paper contains selection of past works related to the problem discussed in this paper.

In the section III, we introduce edge coloring of the set of cubic graphs, proper edge coloring and group of graphs called Snarks (graphs that can't be colored properly with the use of 3 colors).

In the section IV, we present algorithm for edge coloring of the set of cubic graphs and use of permutations of graphs to work with the objectives of this paper.

Computing systems for designed experiments are presented in the section V.

Section VI contains methodologies for experiments with two objectives: explore how does use of computational systems with virtual machines affect slowdown of parallel algorithms which contain recursive functions and search for permutation which would contribute to optimization of time of computation of edge coloring.

Section VII contains experiments and their evaluation. First part of this section consists of experiments which were used to compare systems which use virtual machines with those, which are based on physical machines with the objective of minimizing time of computation of parallel algorithms with recursive functions. Second part of this section is focused on experiments which are used to optimize time of edge coloring of graphs.

II. RELATED WORKS

Our previous work related to using parallel computing systems to solve problem of edge coloring of graphs was focused on the one-graph-at-a-time approach to coloring [2].

Problem of edge coloring of graphs, which is solved and experimentally tested in this paper is related to well-known Four-color theorem presented in [3]. The work of authors of [3], [4], [5], and [6] proves the four-color theorem, that every loopless planar graph admits a vertex coloring with at most four different colors.

Among most significant algorithms used for edge coloring of graphs belong:

- Edge color algorithm by L. Kowalik in [7].
- Heuristic for edge coloring of graphs by authors of [8].
- Edge backtracking algorithm based on breath-first search presented in [9].

Newer works focused on the coloring of graphs are consisting of [10] and [11]. Authors of [10] created population-based memetic algorithm for solving the equitable coloring problem. The equitable coloring problem deals with finding the smallest k for which an equitable legal k -coloring exists. Approach of [11] focuses on computations performed on a set of 35 large benchmark graphs with 450–4000 vertices.

III. EDGE COLORING OF THE SET OF CUBIC GRAPHS

Graph $G(V, E)$ is pair of sets V (vertices) and E (edges). Elements of set E are 2-element subsets of set V [12]. Graphs are cubic when every vertex is incident to three edges, we say that every vertex is of degree equal to 3. Simplest example of cubic graph is presented in the Figure 1.

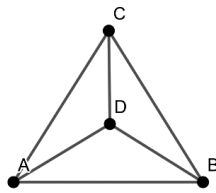


Fig. 1. Example of simple cubic graph

A. Proper edge coloring of cubic graphs

Edge coloring is operation of assigning colors to edges of graph. This coloring is called proper, when there is one instance of each color incident to every vertex. Edge coloring of graph is NP-complete problem [1]. Vizings’ theorem [14] says that minimal number of colors needed for edge coloring of graph G is in the interval $\langle \Delta(G), \Delta(G)+1 \rangle$, where $\Delta(G)$ is maximal degree of vertices in graph G .

Formal notation of Vizings’ theorem is:

$$\Delta(G) \leq \chi'(G) \leq \Delta(G)+1 \tag{1}$$

where $\Delta(G)$ is maximal degree of vertices in graph G and $\chi'(G)$ is edge chromatic index of graph G (minimal number of colors needed for regular edge coloring of graph G) [12].

Value of χ' of most of cubic graphs equals to 3. Those cubic graphs, where the value of $\chi'(G) = 4$ are called Snarks. In the Figure 2, we present two edge colored cubic graphs. One on the left is colored improperly, the other one properly. Graphs presented in this figure has $\chi'(G) = 3$, therefore they are not Snarks. Simplest example of Snark is Petersen graph presented in the Figure 3. It is critical to study this group of

graphs, since for number of important graph problems it is sufficient to prove them for Snarks [15].

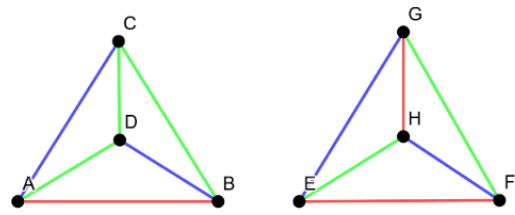


Fig. 2. Example of improperly (L) and properly (R) edge colored cubic graph

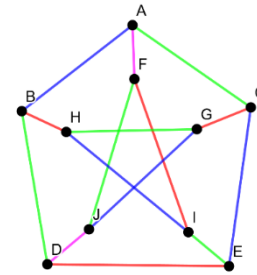


Fig. 3. Petersen graph – simple example of Snark

IV. ALGORITHM FOR EDGE COLORING OF SET OF CUBIC GRAPHS

In the task we need to solve, edge coloring of set of graphs is computed repeatedly up to 10 million times. Therefore, it’s necessary to optimize time of graph coloring computation.

We developed new parallel algorithm for coloring sizable sets of graphs, which calls edge backtracking algorithm multiple times. The edge backtracking algorithm contains recursive functions for edge coloring of graphs. This coloring algorithm cannot be divided (this part of parallel algorithm needs to be computed sequentially). In other words, our parallel algorithm contains sequential parts which cannot be interrupted.

Since coloring part of our algorithm is not parallelable, we need to find some other way of decreasing the time of the computation. Edge backtracking coloring algorithm is coloring edges in order (e.g. first color edge between vertex 1 and 2, then between 2 and 3, and so on). If there are two incident edges colored with same color, algorithm backtracks and re-colors one of the edges.

In the analysis of our previous works we learned that edge backtracking algorithm colors different graphs with same number of vertices and edges in different times. This motivated us to find order of coloring of edges which is colored in shortest time possible. Our new algorithm and methodology are based on examination of time of edge coloring of graph with randomized order of edge coloring. Thus, by permutation of the original graph, we get new sequence of edges for coloring, which might lead to different times of edge coloring of given graph.

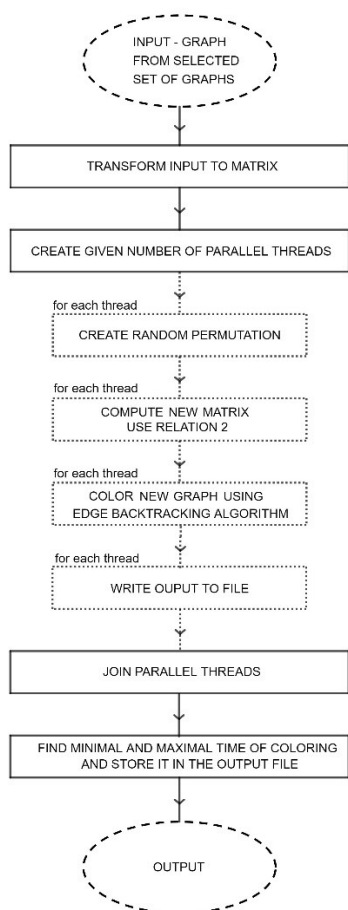


Fig. 4. Simplified schema of algorithm for edge coloring set of cubic graphs

Algorithm in this section is extension of our previous work presented in [2]. Simplified visualization of our algorithm is on the scheme presented in the Figure 4.

As an input for the algorithm we used file containing 19 935 34-vertex Snarks coded in the graph6 format. Algorithm creates random 500 permutations for each one of the input graphs. We generated permutations using conjugation of the adjacency matrix of graph G (see relation 2).

$$A' = P^{-1} * A * P \quad (2)$$

where A is adjacency matrix of given graph G , P is randomly generated permutation matrix (matrix containing exactly one 1 in every row and column), P^{-1} is transposed permutation matrix P and A' is adjacency matrix for new permuted graph G' [2].

We created permuted graphs which have same properties as original graphs but vertices of these graphs are numbered differently. These two graphs (original and permuted) are isomorphic. These permuted graphs are colored in parallel and result of computation is stored in output file – each input graph has its own file containing results of coloring of 500 permuted graphs. Output from the computation of this algorithm consist of:

- time needed for edge coloring of given permuted graph,
- number of recursive calls of the algorithm,

- value TRUE/FALSE which says whether given permuted graph is or isn't colorable with the use of three colors,
- and used permutation in the vector format.

Implementation of our algorithm was done in the C++ while using thread model of parallelization.

V. EFFECTIVE COMPUTING SYSTEMS FOR EDGE COLORING OF SET OF GRAPHS

We are solving the problem of edge coloring of graphs, with use of algorithm which contains recursive function. This motivated us to study use of various computing systems with this type of parallel task. Our aim is to decide whether computing system which uses virtualization is fitting for the problems which are solved with recursive algorithms.

Effectivity of parallel computation is dependent on the selection of fitting parallel computing system. In previous experiments, we encountered technological limitations of computational system. As an example, system we used in the past was not able to compute coloring of approximately 15 graphs in reasonable time. We need to color circa 10 000 000 graphs, which motivated us to use part of high performance cluster.

Even further, it's possible to use computational system with virtual machines, but this type of system can affect duration of computation of parallel algorithm negatively.

We used two computing systems which contain only physical computing resources of High Performance Cluster on Matej Bel University in Banská Bystrica and one computing system which uses virtual machines through the Amazon Web Services cloud environment:

- Physical machine system #1 (PM1) consists of 5 computing nodes (servers) at 16 processors per node, while having 4 GB of RAM.
- Physical machine system #2 (PM2) contains 10 computing nodes (servers) at 12 processors per node and also has 4GB of RAM available.
- Virtual machine system (VM) consists of 4 virtual machines virtualized on Intel Xeon CPU E5-2676 v3 @ 2.4 GHz with 4GB of RAM with the use of Amazon Web Services.

We used these computing systems to measure time of the computation of various sizes of the problem in order to choose the best potential system for computation of the parallel algorithms which contain recursive functions.

VI. METHODOLOGIES FOR TIME OPIMIZATION OF GRAPH COLORING

In previous sections, we established that problem of edge coloring of set of graphs is relevant and it is important to improve time of computing of this problem.

In this section, we present two methodologies based on two main objectives of this paper. Objectives of these methodologies are:

- Use of computing system with virtualization through virtual machines and computing systems with physical machines to computation of the

parallel algorithms which contain recursive functions.

- Search for specific permutation of graph, which can be used for optimization of time of coloring of graphs in the group of Snarks with cyclic 5-edge connectivity. For example, we are looking for such order of edges for coloring, that time of coloring of 34-vertex graphs is lower than 10 ms. We are looking for order of coloring of edges which could be used for optimization of coloring time for set of graphs.

A. Computing systems for edge coloring of set of Snarks

We designed 5 experiments for various sizes of problem of coloring graphs. In these experiments we colored sets of graphs and measured:

- Total time of computation of coloring set of randomly permuted graphs measured by scheduling system of used HPC cluster [16],
- Minimal and maximal number of recursive calls of edge backtracking coloring algorithm measured by our algorithm,
- Minimal and maximal time of coloring of graphs measured by our algorithm.

Small-size problems

We measured these properties for two sizes of problem. First size of the problem is group of small-size problems – this group contains three separate experiments:

- Experiment 1 consist of coloring set of 100 graphs in a ratio 100 input graphs to 1 permutation. This means that we color 1 randomly permuted graph for each of 100 input graphs.
- Experiment 2 contains set of 100 graphs in reversed ratio to experiment 1. The program creates 100 randomly generated permuted graphs of original input graph. These permuted graphs are, then, colored.
- In the Experiment 3, we color 1000 graphs. 1 permuted graph is created for each graph from the set of 1000 input graphs. These permuted graphs are colored.

From the measurements on the small-size problems in the Experiments 1, 2 and 3 we choose computing systems with the result most fitting to work with the other size of problems.

Large-size problems

We used chosen computing systems for the computation of large-size problem group, which consists of two experiments:

- Experiment 4, which contains 2000 input graphs and 500 permutations for each of the input graphs. The set of 1 000 000 graphs is colored.
- In the Experiment 5, used algorithm produces random 500 permutations of each graph from the set of 34-vertex Snarks with cyclic 5-edge connectivity. This creates set of 9 967 500 graphs which is edge colored.

B. Optimization of edge coloring time of Snark set

We designed 3 experiments which work with permutations of graphs in order to optimize time of coloring for selected group of graphs:

- Experiment 6 takes dataset of 9 967 500 graphs (19 935 original graphs, each permuted 500 times) computed in the Experiment 5. From the dataset, we choose time of coloring for all 9 967 500 graphs. We prepare set of data with maximal time of coloring for each one of original 19 935 graphs and similar dataset with minimal times of edge coloring. We visualized and analyzed these datasets. As output from this experiment, we made decision to continue edge coloring graphs with the use of set of 500 permutations with lowest time of coloring.
- In the experiment 7, we change coloring of randomly generated permuted graphs to coloring graphs with chosen set of permutations applied on them (see relation 2). This set is consisting of 500 permutations with lowest time of coloring measured in the experiment 6.
- Analogically to experiment 7, in the experiment 8 we change coloring of randomly permuted graphs to coloring graphs permuted with the use of permutation with lowest time of coloring applied to the original set of Snarks.

VII. EXPERIMENTS & EVALUATION OF METHODOLOGIES

Based on methodology described in the section VI, we present experiments related to both methodology objectives.

Subsection A presents experiments and their evaluation with the objective to compare computing systems which use physical machines and systems which use virtualization. As part of these experiments, we are looking at slowdown of parallel algorithm which uses recursive functions in the system which uses virtualization.

Subsection B contains experiments and evaluation of search for specific permutations, which optimize time of coloring of graphs in the group of Snarks with cyclic 5-edge connectivity

A. Experiments – Computing systems for edge coloring of set of Snarks

Experiment 1

TABLE I. EXPERIMENT 1 - 100 GRAPHS, 1 PERMUTATION

	PM1	PM2	VM
Overall time [sec]	< 1	< 1	< 1
CPU time [sec]	3	5	1
Minimal number of recursive calls	32	33	1001
Maximal number of recursive calls	3 139 482	3 291 503	28 977
Minimal time of coloring [ms]	< 1	< 1	< 1
Maximal time of coloring [ms]	440	530	16

In the table I, we present results of measurements from experiment which consist of coloring set of 100 graphs. We color 1 randomly permuted graph for each of 100 input graphs.

Experiment 2

This experiment contains set of 100 graph. The algorithm is given one input graph and it creates 100 randomly generated permutations for edge coloring. Measurements from this experiment are presented in the table II.

TABLE II. EXPERIMENT 2 - 1 GRAPH, 100 PERMUTATIONS

	PM1	PM2	VM
Overall time [sec]	< 1	< 1	< 1
CPU time [sec]	2	3	1
Minimal number of recursive calls	23	32	672
Maximal number of recursive calls	3 328 041	3 211 076	40 619
Minimal time of coloring [ms]	< 1	< 1	< 1
Maximal time of coloring [ms]	540	610	5

Experiment 3

Experiment 3 has 1000 input graphs. 1 permuted graph for each of the set of 1000 input graphs is colored. Table III presents all measured properties of the run of program.

TABLE III. EXPERIMENT 3 - 1000 GRAPHS, 1 PERMUTATION

	PM1	PM2	VM
Overall time [sec]	< 1	< 1	< 1
CPU time [sec]	15	16	3
Minimal number of recursive calls	5	33	710
Maximal number of recursive calls	2 511 002	3 501 601	37 210
Minimal time of coloring [ms]	< 1	< 1	< 1
Maximal time of coloring [ms]	330	480	9

Systems we were using for these experiments were not able to work with the size of problem consisting of 1 graph on the input and 1000 permutations colored in parallel.

From experiments 1, 2 and 3 on the small-size problems we chose architectures PM1 and VM. We used these architectures to compute large-size of problems presented in the experiments 4 and 5.

Experiment 4

We use our algorithm to color set of 1 000 000 graphs. Ratio of this set is 2000 input graphs to 500 permutations. This means that we edge color 500 permuted graphs for each element of input set of graphs.

In the table IV, we can see, that overall time of run of our application was much shorter on the VM architecture than on PM1.

TABLE IV. EXPERIMENT 4 - 2000 GRAPHS, 500 PERMUTATIONS

	PM1	VM
Overall time [min]	34:59	2:07
CPU time [min]	44:58	15:10
Minimal number of recursive calls	42	10
Maximal number of recursive calls	9 693 080	1 000 780
Minimal time of coloring [ms]	< 1	< 1
Maximal time of coloring [ms]	1280	18

Experiment 5

In the table V, we present measurements from experiment 5, in which we color 500 permutations of each graph from the set of 34-vertex Snarks with cyclic 5-edge connectivity. This creates the set of 9 967 500 graphs.

On the contrary to experiment 4, we observe, that overall times of computation of the problem were similar for both architecture PMA1 and VMA.

TABLE V. EXPERIMENT 5 - 19 935 GRAPHS, 500 PERMUTATIONS

	PM1	VM
Overall time [h]	2:41:13	2:32:35
CPU time [h]	4:53:38	3:00:54
Minimal number of recursive calls	31	548
Maximal number of recursive calls	1 283 104	866 762
Minimal time of coloring [ms]	< 1	< 1
Maximal time of coloring [ms]	1260	29

Evaluation of experiments on computing systems for parallel algorithms which contain recursive functions

Use of virtualization in the computations related to coloring of graphs with parallel algorithm which contains recursive function doesn't seem to create slowdown or latency of any kind. In some cases (experiments 3 and 4), it was much more effective to work with computing system which used virtualization through virtual machines, than to use physical machine system.

Use of parallel computing systems in combination with parallel programming for finding solutions to graph coloring problems is important. We were able to shorten the time of coloring of the set of 34-vertex Snarks with cyclic 5-edge connectivity on physical machines from 17 hours and 36 minutes to 2 hours and 41 minutes.

B. Experiments – Edge coloring time optimization for set of Snarks

Experiment 6

In the experiment 6 we present maximal time of coloring from all 500 permutations of each of the original 19 935 graphs.

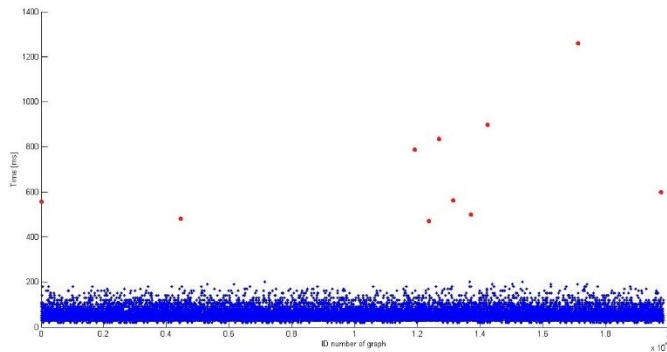


Fig. 5. Permutations with the maximal time of coloring for each of the original 19 935 graphs

Figure 5 shows, that maximal time of coloring of permuted graphs was largely grouped between 20 and 200 ms. In the red, we highlighted permuted graphs with extremely long time of coloring.

Same as permutations with maximal time of coloring, we created set of 19 935 permutations with minimal time of coloring. All elements of this set are permutations with the time of coloring under 1 ms.

Experiment 7

In the experiment 7, we switched from generating 500 random permutations to targeting the set of permutations with minimal time of coloring. Maximal times of coloring for the set of 500 chosen permutations applied on the original set of 19 935 snarks are presented in the Figure 6.

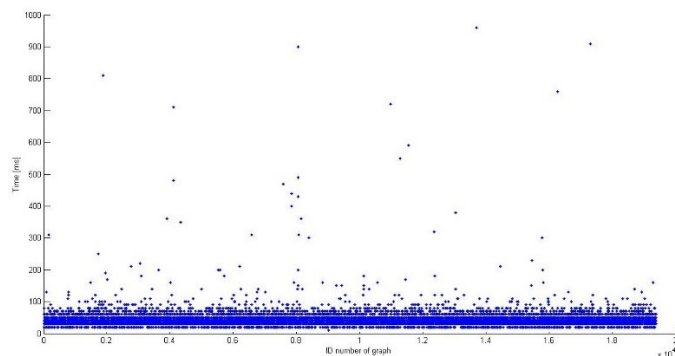


Fig. 6. Maximal times of coloring for chosen 500 best permutations applied on the of 19 935 Snarks

Figure 6 shows two decreases in the maximal times of coloring:

- Main group of graphs is colored in the times between 20 and 100 ms, this range was previously between 20 ms and 200 ms.
- Range of extremely high times of coloring is decreased from 1260 ms to 960 ms.

Experiment 8

From the set of 500 permutations with lowest times of coloring, we chose one with best time. We applied this permutation on the original set of 19 935 graphs and colored them. Figure 7 presents decrease of times of coloring to values lower than 0,2 ms for main group of graphs with extremes in the 10 ms range.

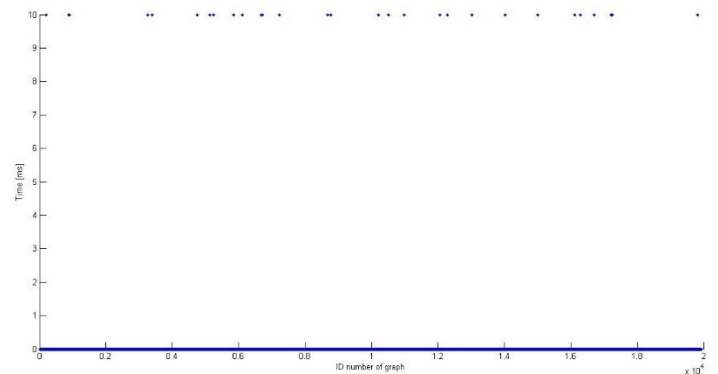


Fig. 7. Maximal times of coloring for best permutation applied on the of 19 935 Snarks

Evaluation of optimization of edge coloring time of set of Snarks

In the experiments 6 and 7, we can recognize different decreases in maximal times of coloring. Edge backtracking coloring algorithm is sensitive to the initial edge of coloring on the group of Snarks, we were working with in this paper.

In the experiment number 8, we found permutation applicable on graphs from selected group, which lowered maximal times of coloring from the initial interval of 20 – 1260 ms to 0,2 – 10 ms.

VIII. CONCLUSION

This paper is motivated by the need for optimization of time of edge coloring of graph sets. Even though, we were focused on edge coloring of Snarks, edge coloring of any graph is NP-complete problem which invites use of parallel computing systems and specialized algorithms.

We developed new parallel algorithm usable for coloring of any set of graphs. We also designed, implemented and presented methodology used for specification of parallel computing system, which is practical for computation of graph coloring related problems and methodology usable for time optimization of chosen task – edge coloring of 34-vertex Snarks with cyclic 5-edge connectivity.

Regarding use of high performance computing in the research of selected group of Snarks, we conclude four main findings:

- Use of parallel computing systems and parallel programming for finding solutions to graph coloring problems is crucial. We were able to improve (decrease) time of coloring of chosen set of graphs on physical computing system from 17,5 hours to 2 hours and 40 minutes.

- Use of virtualization in computing system for the computations with recursive functions doesn't seem to create slowdown or latency of any kind. In some cases (on small size problems), it is more effective to work with computing system which uses virtualization through virtual machines, than to use physical machine system.
- As shown in the experiments 6 and 7, edge backtracking coloring algorithm is sensitive to the order of edges on the group of Snarks, we were working with in this paper.
- On the chosen group of graphs, it is possible to find successions of vertices (permutation) that causes short time of coloring of this set of graphs.

We are currently working on an understanding of which, if some, properties of the established permutation(s) can be generalized for group of all 34-vertex Snarks and, later, for all Snarks.

ACKNOWLEDGMENT

Computing was performed in the High Performance Computing Center of the Matej Bel University in Banská Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

The research was partially supported by the grant of The Ministry of Education, Science, Research and Sport of the Slovak Republic, VEGA 1/0487/17.

REFERENCES

- [1] I. Holyer, "The NP-Completeness of Edge-Colouring," in SIAM J. COMPUT, vol. 10, Issue 4, 1981, pp. 718-720.
- [2] A. Dudáš, P. Voštinár, J. Škrinárová, J. Siláči. (2018) "Improved process of running tasks in the high performance computing system". Proceedings of 16th IEEE International Conference on Emerging eLearning Technologies and Applications. New Jersey: Institute of Electrical and Electronics Engineers. ISBN 978-1-5386-7912-8. pp. 133-140
- [3] N. Robertson, D. Sanders, P. Seymour, R. Thomas. "The Four-Colour Theorem". Journal of Combinatorial Theory, Series B, Volume 70, Issue 1, 1997, Pages 2-44, ISSN 0095-8956
- [4] K. Appel, W. Haken, "Every Planar Map Is Four Colorable", A.M.S. Contemp. Math. 98 (1989).
- [5] K. Appel, W. Haken, "Every planar map is 4 colorable. Part I". Discharging, Illinois J. Math. 21 (1977), 429-490.
- [6] K. Appel, W. Haken, J. Koch, "Every planar map is four colorable. Part II." Reducibility, Illinois J. Math. 21 (1977), 491-567.
- [7] L. Kowalik, "Improved Edge Coloring with Three Colors", Theoretical Computer Science, Vol. 410, No. 38-40, 2009, 3733-3742.
- [8] M.A. Fiol, J. Vilaltella, "A simple and fast heuristic algorithm for edge-coloring of graphs", arXiv:1210.5176v1 [math.CO]
- [9] R. Nedela, J. Karabáš, M. Škoviera, "Nullstellensatz and Recognition of Snarks". 52th Czech-Slovak Conference Grafy, 2017
- [10] W.Sun, J.-K.Hao, W.Wang et al., "Memetic search for the equitable coloring problem", Knowledge-Based Systems (2019) 105000
- [11] Q. Wu, Q. Zhou, Y. Jin, J. Hao. "Minimum sum coloring for large graphs with extraction and backward expansion search". Applied Soft Computing, Volume 62, 2018, Pages 1056-1065, ISSN 1568-4946
- [12] R. Diestel, *Graph theory*. 5th edition. Springer - Verlag, Heidelberg, 2016
- [13] G. Brinkmann, K. Coolsaet, J. Goedgebeur, H. Mélot. "House of Graphs: a database of interesting graphs". Discrete Applied Mathematics, 161:311-314, 2013 (DOI). Available at <http://hog.grinvin.org>
- [14] Vizing, V. G. (1964), "On an estimate of the chromatic class of a p-graph", *Diskret. Analiz.*, 3: 25-30, MR 0180505
- [15] G. Brinkmann, J. Goedgebeur, J. Hagglund and K. Markstrom, "Generation and properties of Snarks", Journal of Combinatorial Theory, Series B, 103(4):468-488, 20137
- [16] PBS professional. PBS professional: Industry-leading workload manager and jobscheduler for high-performance computing. [Online]. Available: <https://www.pbspro.org/>. [Accessed: september, 2019]
- [17] T. He, A. N. Toosi, R. Buyya. "Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers". Journal of Parallel and Distributed Computing, Volume 131, 2019, Pages 55-68, ISSN 0743-7315

Pre-diagnostic of diabetes with the use of statistical machine learning

Jana Dudova
Faculty of Management Science
and Informatics
University of Zilina
Zilina, Slovakia
dudova5@stud.uniza.sk

Emil Krsak
Faculty of Management Science
and Informatics
University of Zilina
Zilina, Slovakia
emil.krsak@fri.uniza.sk

Tomas Kello
Faculty of Management Science
and Informatics
University of Zilina
Zilina, Slovakia
tomas.kello@fri.uniza.sk

Abstract—The aim of this work is to make a prediction of diabetes, which included acquiring knowledge about diabetes, their habits, food income, daily activity. Processing data, producing a statistical model and implementing the model into an application. Which can predict the probability of being a diabetic, based of his lifestyle and genetic assumptions.

Index Terms—Logistic regression, machine learning, R language, data mining

I. INTRODUCTION

Modern times offer a large amount of data that can be processed and also easy to obtain. Individual data offers a wide range of information and facts. New data is added every day and grow exponentially. We will extract data related to diabetic to improve results.

Diabetes mellitus is a disease that currently has nearly 400 million people. Scientists assume that this number will increase and, unfortunately, it will be quick. If diabetes mellitus is not detected or controlled in the patient, it may cause complications incompatible with life. Along with this disease, cardiovascular disease develops more. Diabetes is a serious threat to public health worldwide.

In today's digital age, there is still a large group of people who haven't knowledge about diabetes mellitus as well as about overall health. We think, it is important to spread medical education to all parts of the world and to help improve diagnostic methods in this area. Early diagnosis and prevention can save many lives. In preventing the disease, it is important to keep your body moving, monitor your body weight and generally maintain a healthy diet.

There is great potential for statistical learning in the health and prediction of certain diseases that we have used in our work. The data obtained and their proper processing can get a result that can be helpful to the doctor in first contact with the patient. On the other hand, patients can test their health from their homes with feedback that can help doctors. The above-mentioned potential was the reason why we chose this topic. [1]

II. METHODS

The logistic regression allows data analysis, where the response can be a binary or frequency variable expressed by

0 or 1. Alternatively, a ratio in the range $< 0, 1 >$. The ratio represents the proportion of positive results for multiple repeat measurements at the same value of the independent variable x . The number of measurements should be approximately the same for each ratio. Typical applications include technology estimation of failure or failure risk under certain conditions. In finance, this may be a prediction of the risk of credit, depending on economic indicators. The probability of death in ecology and biology. Logistic regression is also used in marketing to predict the client's transition to competition, but in recent years, we have seen the incidence of logistical regression in pharmacy and medicine to a significant extent. [2]

Considering a binary variable Y , which obtains a value of 1 with the probability π and a probability $(1 - \pi)$ of 0. It represents a vector:

$$X'_i = [x_{i1}, x_{i2}, \dots, x_{ik}], i = 1, 2, \dots, n, \quad (1)$$

i -th combination of values to non-random explanatory variables X_1, X_2, \dots, X_k , i -th conditional distribution of Y is an alternative with π_i and probability:

$$P(y_i | \pi_i) = \pi_i^{y_i} (1 - \pi_i)^{1 - y_i} \quad (2)$$

The regression function with logit π could be explained as:

$$g(\pi) = \ln\left(\frac{\pi}{1 - \pi}\right) = x' \beta, \quad (3)$$

where

$$x' = [1, x_1, x_2, \dots, x_k],$$

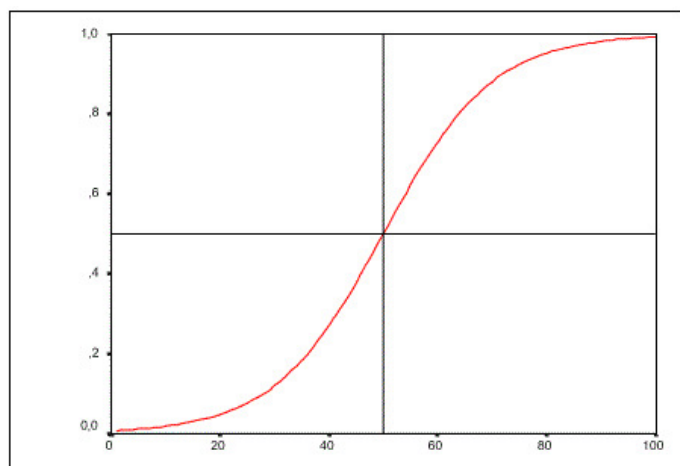
$$\beta = [\beta_0, \beta_1, \dots, \beta_k],$$

This is called a logistic regression function. A conditional mean value with a binary variable can be expressed as a nonlinear function of the explained variables. From equation 3 it follows that:

$$\frac{\pi}{1 - \pi} = e^{x' \beta}, \quad (4)$$

and on

$$\pi = (e^{x' \beta}) / (1 + e^{x' \beta}) = [1 + e^{-x' \beta}]^{-1} \quad (5)$$

Figure 1. Graph of π when $k = 1$

An example of a function where $k = 1$ from equation 3 can be seen in the figure 1.

Since the equation 5 belongs to the distribution function of the logistic distribution, we can say that the linear combination of the explanatory variables $x'\beta$ (logit) is 100π - percentile quantile.

The bounding of values to the interval $\langle 0, 1 \rangle$ is obtained by using the distribution function of distribution for modeling the probability π , which the given value limitation guarantees to us. We also know other frequently used distributions in which the distribution function in the graph is plotted as a symmetric S-curve. It is a normal distribution where a linear combination of explanatory variables can be expressed by its 100π percentile, otherwise called probit. In statistics, the use of logit models is preferred, although the probability estimation is very similar in both cases. Logit model is easier to interpret.

Logistic regression and its function is an explanatory variables. There are categorical variables and numerical continuous variables. Knowing the nature of these explanatory variables is important in model estimation, construction, as well as in determining the quality of a given model or its interpretation of parameters. Based on these facts, it is also possible to use it in the future.

A. Other data mining tools

Data mining can be imagined as data collection and also as a data evaluation process. This term is often associated with the Business Intelligence industry, whose main task is to analyze data and work with business information.

Decision trees: belong to a group of classification models. By the decision tree, we can imagine a tree where each node that is non-leafy represents an attribute test at its value. [3]

They are made up of a group of hierarchically arranged rules for decision making. It consists of a root that represents the entire set, followed by the branching of other nodes. We say the tree is growing. Nodes that no longer follow branching are called leaves. According to the type of dependent variable, the trees are divided into a regression and classification tree.

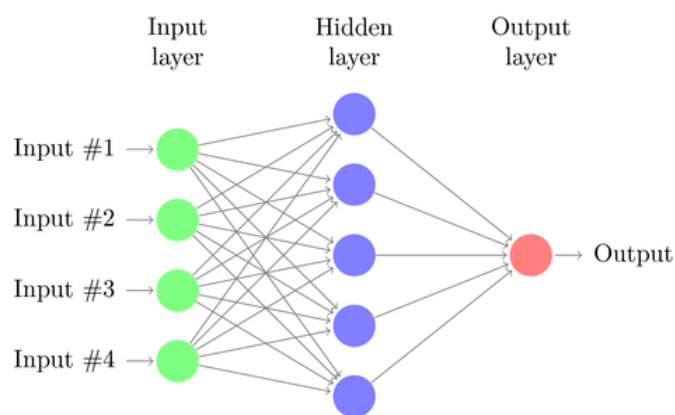


Figure 2. Neural network

Neural networks: is based on algorithm whose model is human brain and its activity. The human brain is made up of a tangle of neurons that communicate with each other through impulses. The neural network is part of the data mining that has already been mentioned. The essence of neural networks is now implemented in countless analytically and decision-making software. Also in the field of human competence. In all the directions that the neural network receives, it shows excellent results by comparing standard decision algorithms. [4]

The principle of neural networks, as their name predicts is similar to the biological neural network. The basic unit is the nerve cell, the neuron. Neurons are interconnected as shown on figure 2 and at the same time have the ability to adapt on the basis of the training data set. In data analysis, neural networks give a wide range of new possibilities. The biggest advantage of neural networks is their ability to learn. They make use of the memories of the combinations we have made, which have led us to the aforementioned result. Subsequent use of already acquired experience and learned combinations to solve other problems. The benefits of neural networks include the notion of generalization. It is the ability to respond to inputs that were not contained in the training set and to prove the outcome or to draw conclusions about the issue. Learning predisposition for neural networks is also considered a definition of artificial intelligence.

B. Hypothesis testing

If we only have data that is obtained from a random sample and want to express a certain statement about the main file, then in these cases we use inductive statistics. In our case, the data for the classification model will be measured. We use induction to learn and make judgments. However, care should be taken when creating inductive considerations that do not always imply the correctness of the judgment. Inductive statistics seek to use their methods to minimize misconception. The basic task of the above-mentioned statistics is to estimate parameters and test them. In inductive statistics we distinguish two basic types of parameter estimates.

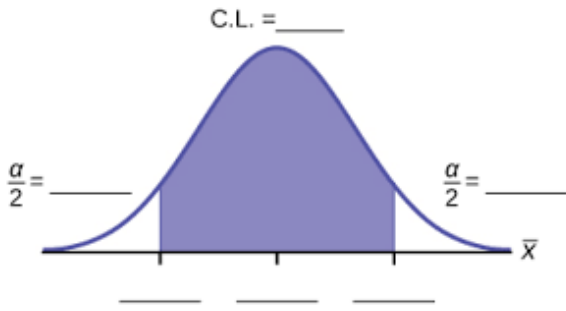


Figure 3. Confidence interval

The first type is called point estimates. Considering that we have a random selection (X_1, X_2, \dots, X_n) of a given distribution that is described by the distribution function and we know that the value of the distribution function varies depending on the selection of the real number and we also classify θ into the characteristics of the distribution, then the task of the point estimate is to find T_n equal to $T(X_1, X_2, \dots, X_n)$, which takes values related to that unknown parameter θ . It is called $\hat{\theta}$. [5]

There are three estimation properties:

- 1) Consistency - T_n is a consistent estimate of θ if:

$$\lim_{n \rightarrow \infty} P(|T_n - \theta| < \epsilon) = 1, \text{ for } \epsilon > 0 \quad (6)$$

- 2) Efficiency - T_n is an effective estimate of θ if it has the smallest dispersion among all estimates
- 3) Non biased - if $E(T_n) = \theta$. We can argue that T_n is an unfavorable estimate of θ , but it does not guarantee that the estimate is correct. This is obtained by asymptotically deflected estimation of the parameter θ if:

$$\lim_{n \rightarrow \infty} E(T_n) = \theta \quad (7)$$

In interval parameter estimates, we look for a value between a given interval, where we can say with certainty that the searched value is in the interval. As with the above-mentioned point estimation, the interval estimates of parameters will be based on a selection made randomly. We will look for an interval in which θ will be found with probability $1 - \alpha$, where $1 - \alpha$ denotes reliability and α risk.

The procedure to build a confidence interval that can be both one-sided and double-sided can be summarized as follows

- 1) Find the point estimate T_n of parameter θ .
- 2) We will try to find the function $t(T_n, \theta)$, which has a known probability distribution.
- 3) Find t_1, t_2 as: $P(t_1 < t(T_n, \theta) < t_2) = 1 - \alpha$, where t_1, t_2 are the critical distribution values $t(T_n, \theta)$.
- 4) Adjust $P(t_1 < t(T_n, \theta) < t_2) = 1 - \alpha$ to form $P(T_D < \theta < T_H) = 1 - \alpha$ in the case of two-sided confidence interval, where T_D is the lower bound of the confidence interval and T_H is the upper limit of the confidence interval.

For the right-hand confidence interval: $P(\theta < T_H) = 1 - \alpha$ and for the left-hand confidence interval: $P(T_D < \theta) = 1 - \alpha$.

Approaches to hypothesis testing: when testing hypotheses we use parametric tests, where we need to know the distribution of random variables and non-parametric tests, where we do not know about the type of distribution.

H_0 usually indicates a hypothesis that we are trying to verify. It is called a test or null hypothesis. The null hypothesis can be imagined as equality between the hypothetical parameter value and the unknown parameter value. It should be remembered that the null hypothesis is always tested. If we already have a formulated test hypothesis together with random selection, we will try to define the alternative hypothesis H_a . If H_0 equals an unknown parameter value with a hypothetical parameter value, H_a can be written as follows:

$$H_a = \theta = \theta_1$$

simple alternative

$$H_a = \theta \neq \theta_0$$

two-way alternative

$$H_a = \theta < \theta_0$$

one-sided alternative

$$H_a = \theta > \theta_0$$

one-sided alternative

(8)

Testing statistical hypotheses has two approaches. First, we will clarify the first, classical procedure for testing hypotheses. First, we determine H_0 and H_a . When we define both hypotheses, we add to them the significance level α , which is in the range of 0 to 1. The definition of the critical field and the definition of the statistics

$$T = T(X_1, X_1, \dots, X_n; \theta), \quad (9)$$

where T is criterion. If the test criterion value lies in the critical region, I reject H_0 in favor of H_a . If the test criterion does not belong to the critical domain, H_0 does not reject the significance level α .

The second hypothesis testing option is a pure significance test. In this test, in addition to defining zero and alternative hypothesis, the significance level α is looking for the value of the test criterion p -value. P -value is the calculated value, which indicates the lowest level of significance where we can reject the null hypothesis. When evaluating, we compare the already mentioned p -value with the significance level α . If p -value is less than or equal to α , H_0 is denied on the significance level α . If p -value is greater than α , H_0 is not denied at the significance level α . The significance level is the most often take the following values: 0.1, 0.05, 0.01 and 0.001. P -value graph is shown on figure 4.

C. Data processing

Data imputation: guarantees the process of replacing missing data of one or more variables with a substituted value.

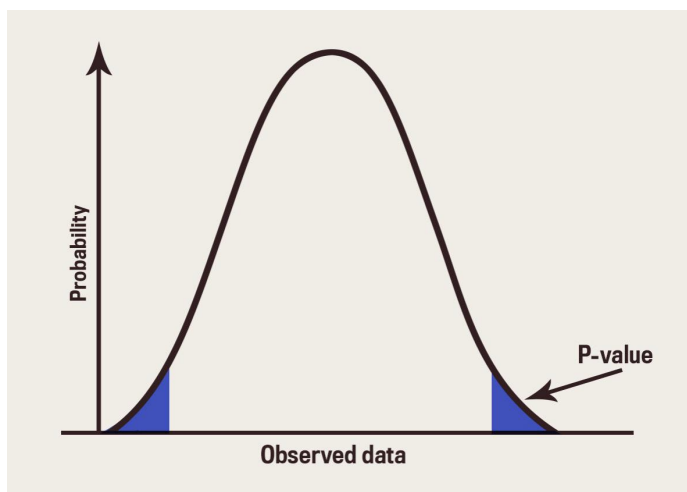


Figure 4. P-value

In general, we know more ways of imputation that solve the problem of missing data.

We will learn two ways of one-dimensional imputation of missing data. The first is the deductive imputation that we will now approach. Deductive imputation is the deduction of missing values based on derived logical connections and relationships between variables. This means that if we can calculate the missing value, we will do it anyway. This method is one of the simplest methods of imputation.

The second method is to replace the average value. This method is more challenging than the first method. However, as its name suggests, the missing data is replaced by a value that indicates the arithmetic mean of the observed values of the respective variable. With this replacement procedure, we encounter the fact that all values that are missing the appropriate variable will be replaced by the same value one by one. This means that adding a certain value to the dataset will only increase the size of our sample and not get new information. Despite the problem, this method of imputation is among the more popular.

1) *Correlation*: is necessary to find out how strong the relationship is between the variables. As a dependent variable, we chose whether diabetes is diagnosed, assuming we are making prediction of diabetes. The variable is represented by the name *DiagCukrovka*. It gets values "0" as no and "1" as yes. Because there are categorical variables in our model, we use The GoodmanKruskal package to determine the dependency. If they are only integer, we can use a classic correlation. [6]

Goodman-Kruskal is founded on random category assignment. It measures for cross tabulations of nominal level variables and also measures the percentage improvement in predictability of the dependent variable given the value of other variables.

		The Truth		
		Has the disease	Does not have the disease	
Test Score:	Positive	True Positives (TP) a	False Positives (FP) b	$PPV = \frac{TP}{TP + FP}$
	Negative	False Negatives (FN) c	True Negatives (TN) d	
		Sensitivity	Specificity	
		$\frac{TP}{TP + FN}$	$\frac{TN}{TN + FP}$	
		Or, $\frac{a}{a + c}$	$\frac{d}{d + b}$	

Figure 5. Sensitivity and specificity

D. Model of logistic regression

After quantifying the model, we decided to determine the overall success of the trained model as well as the specificity and sensitivity values. Sensitivity is the ability of a model to recognize really sick people. This means that the person who is sick will belong to a set of positive results. Specificity is the ability of a model to recognize really healthy people who are not sick. This means that they will belong to a set of negative results.

In the calculation, we experimented with a t-value or otherwise called a threshold value. This value can take a value from 0 to 1. Most often t - value is set to 0.5, thanks to the possibility to change these values we can adapt the model to requirements of different kinds. An example is the high threshold value, where it is possible to detect people with a very high risk of getting diabetes.

The ROC (Receiver Operating Characteristics) graph is the second way to find a suitable t-value. For a better understanding of the curve, we have to say that the ROC curve is based on a table of correct or incorrect prediction. First we used experimentation and later mentioned graph. For the ROC curve, we have to look for a compromise, which means that the ratio of real positives is as high as possible, while the false positives ratio is as low as possible. Our ROC graph is shown on figure 6. According to this curve, we can see what change the size of the t-value makes. T - value from 0.4 - 0.5, 0.6 - 0.7 and 0.8 - 0.9 will significantly improve the model prediction. We can see this on a curve that literally jumps up at these values. The area under the curve is one of the criteria for selecting a suitable and quality model. For us, the area under the curve was 0.832. We are satisfied with the value.

The result of the prediction from the logistic regression model is obtained by the following formula:

$$P = \frac{e^{(a+bX)}}{1 + e^{(a+bX)}} \quad (10)$$

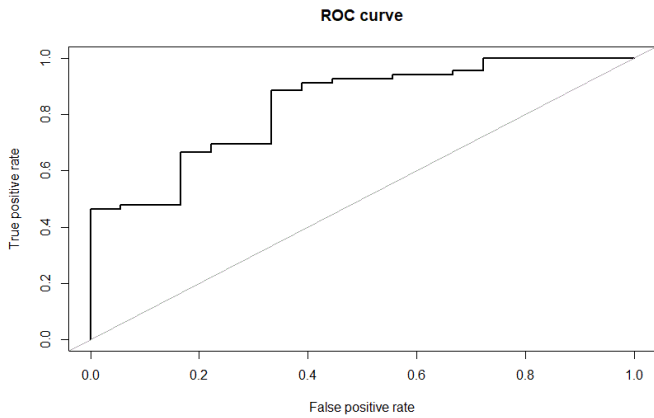


Figure 6. Receiver Operating Characteristics curve

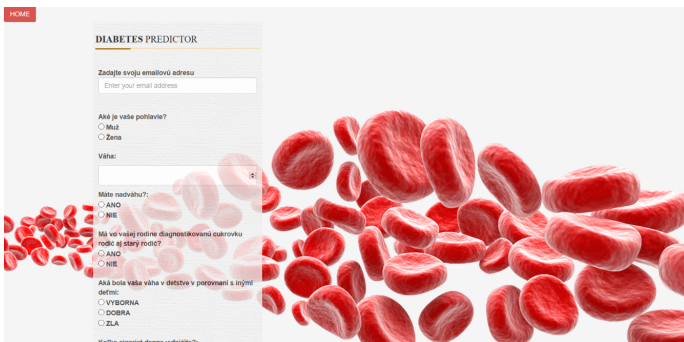


Figure 7. Web quiz

E. Implementation of the statistical model

To implement a statistical model into an application, we chose to apply it as a web page, where we can find a quiz to calculate the likelihood of getting a diabetes from a given respondent.

The web-based quiz contained 22 different questions from sections such as physical, marital and education status or health. The quiz was created on the basis of a questionnaire filled in by people suffering from diabetes mellitus.

Subsequently, we used test data. From the above questions, we used the recorded values to calculate probability, which were statistically significant by logistic regression. These included weight, overweight, cigarette smoking, alcohol use, family diabetes or childhood weight. We used the other recorded data to process statistics about the test.

III. RESULTS

The results that were achieved and the essence of our research was the creation of diabetes prediction. The implementation in the form of a web application was an attempt to bring our results closer to the public, to users of different age groups and also to different technical education.

We have discussed our work with ordinary Internet users. Their suggestions and insights have tried to include in the result. The effort in the practical part of the thesis was a

Gender	What is your gender?
Weight	What is your weight
BloodPressTab	Are you taking blood pressure tablets
Diabete	Diabetes diagnosed?
DiabeteFamily	Do you have diabetes in family
Preiabete	Prediabetes diagnosed?
Overweight	Overweight diagnosed?
WeightKid	What was your weight during childhood?
WeightNow	Are you satisfied with your current weight?
EatOut	How many times have you eaten in restaurant last week?
Allergy	What allergy do you have?
Cholesterol	You have elevated cholesterol?
RegCheckUp	Do you take a regular preventive check-up?
Fruits	How often do you eat fruit?
FastFood	Are you eating fast food?
Salt	How salty food are you eating?
Sweet	How sweet food are you eating
Breakfast	How many times did you eat breakfast this week?
BreakfastKid	Did you eat breakfast when you were a kid?
SleepLength	What is your sleep length?
ActivityHours	How many hours do you do some activity per week?
WorkHours	How many hours do you work per week?
Smoking	How many cigarettes do you smoke per day?
Alcohol	How often do you drink alcohol?
Coffe	How many coffe do you dring per day?
Stress	Are you in stress during a day?

Table I
TEST QUESTIONS

vision of the implementation the model into practice. However, with perceived perceptions and necessary tuning, this is not immediately possible.

The results of the self-created questionnaire and the subsequent creation of a statistical model showed that the factors influencing diabetes are weight, diagnosed overweight, alcohol and cigarette use, diagnosis of diabetes in the family and comparison of respondents weight in their childhood to other children. This means we have a different weight than most children in our neighborhood.

When we compare the opinion on diabetes, which it shows on the basis of data obtained from all over the world, that overweight and poor living lead to an increased likelihood of getting diabetes, we came to the same conclusion. The best prevention of the disease is sufficient movement, keeping the right body weight and last but not least, the right diet. Unfortunately, we do not know how to influence genetically determined factors, but we can delay the onset of the disease.

The beginnings of our work go back to getting data that has not always been successful. Our aim was to obtain a proven data set with a wide range of respondents. Unfortunately, unsuccessfully. The second alternative was to get data from a self-made questionnaire that we distributed through social networks in communities and groups of people who have diabetes mellitus. It was also necessary to obtain data from healthy people. After obtaining and evaluating the collected data, we continued to build the model.

We created the model in the R language and R studio, which is designed to work with a statistical programming language. The result of the model creation was a logistic regression model. Prior to the final model, the loaded .csv file

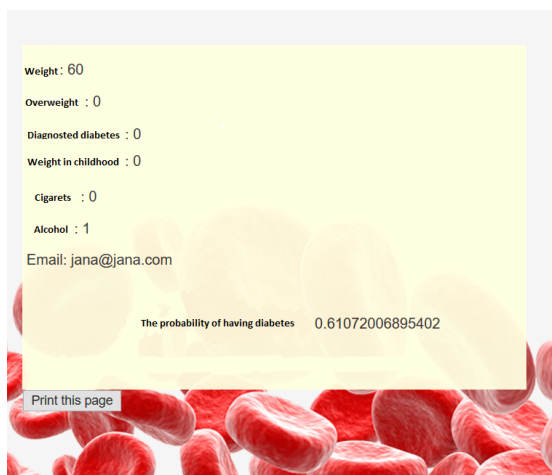


Figure 8. Results of Web quiz

had to be edited for R. The modifications included imputation, which means the replacement of the missing values and the correlation that serves to determine the degree of dependence between the variables. We used the final modified data set to create the aforementioned regression model, which resulted in a fulfilled model consisting of those variables that were statistically significant to it.

It was necessary to test the filled model on both, the training and test set, which we divided before the model was created. It was important to determine the overall success on both sets, which should be correct after its values. Then we compared the results of the decision tree, which was in favor of logistic regression, and we confirmed it with the ROC curve.

Gradually, we went to create a practical part, which was to create a web application. Our web application has been divided into a user section and an admin section. The main essence was a questionnaire, which calculated the probability of obtaining said diabetes on the basis of statistically significant variables from the logistic regression model. The entered result was stored in the database together with the user's email address. Admin pages had the option to look into this database after logging in and also send an email to that user. The web application also offered some additional features, such as website management or graphical outputs from stored results in the database.

IV. CONCLUSION

In our work we collected data from different people and created data set. Next step was to analyze them, fill missing data and divide them to train and test group. We worked with language R and conclude that the use of data and language R can bring in the future many improvements in the diagnosis of diseases. However, it is necessary to add that awareness of the public with the mention of the R language and the use of data is necessary to improve, to make better use of the maximum possibilities and capabilities of the offered sphere.

As we can see, data collection and processing is great importance and a great deal of influence in decision making

in multiple spheres. It is up to us how we use the data. We believe that in the future the prediction of the disease in this form will be a common phenomenon and will help patients live a longer and full life.

ACKNOWLEDGMENT

The authors would like to thank to the anonymous referees for their valuable answers about their life and personal data. They helped us with creating data set to work on it, which also improved quality of the paper.

REFERENCES

- [1] G. M. Reaven and E. Reaven, "Diabetes mellitus," in *Nutritional Aspects of Aging: Volume 2*, 2018.
- [2] H. Schwender and I. Ruczynski, *Logic Regression and Its Extensions*. Elsevier Inc, 2010.
- [3] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, 1986.
- [4] J. Schmidhuber, "Deep Learning in neural networks: An overview," 2015.
- [5] L. L. Johnson, C. B. Borkowf, and P. A. Shaw, "Hypothesis Testing," in *Principles and Practice of Clinical Research*, 2017.
- [6] T. Speed, "A correlation for the 21st century," 2011.

Software architecture of Automatic Train Operation

Michal Ďuračik*, Emil Kršák*, Matej Meško* and Ján Ružbarský*

* University of Zilina, Faculty of Management Science and Informatics, Univerzitná 8215/1, 010 26 Zilina Slovakia
E-mail: Michal.Duracik@fri.uniza.sk

Abstract—Nowadays, systems are being created that can replace (fully or partially) human train control with artificial. Such problematic is included in the project “ATO over ETCS” (ATO). The aim of this project is to develop a pilot information system that will guide train in consideration of travel schedule in real-time. There are two main parts of ATO: unit located in the train and server part. Parts use the *UNISIG* protocol for communication which was specially designed for the needs of the project. We design architecture, services and their individual components of the server part. Also, we make an analysis of systems used in a railway environment and its data sources, design communication between such systems, design algorithms for this data processing so they can be used in ATO communication. Actually, the project is in the phase of laboratory tests. The next phase will be the verification of ATO in real-world railway track.

Keywords—information system, automatization, control, communication

I. INTRODUCTION

Within Europe, the *Shift2Rail* association of companies was founded. Aim of the association is the development of innovative solutions for railways. One of its projects is Autonomous Train Operation over *ETCS* (ATO). Its main goal is to design, develop and demonstrate on the real railway track system, that can manage trains in real-time exactly according to their schedules and fulfill all security restrictions. System specification of ATO security level is semi-automatic security level *GoA2* [1]. The train is autonomous until it gets actual relevant data. In case of data issues, the control is left to train driver. ATO must cooperate with all his sources: *ETCS* (European Train Control System) and *TMS* (Traffic Management Systems) [2].

In the final stage of the project, we can test such a system on a real train that will go e.g. from Austria to Slovakia autonomously.

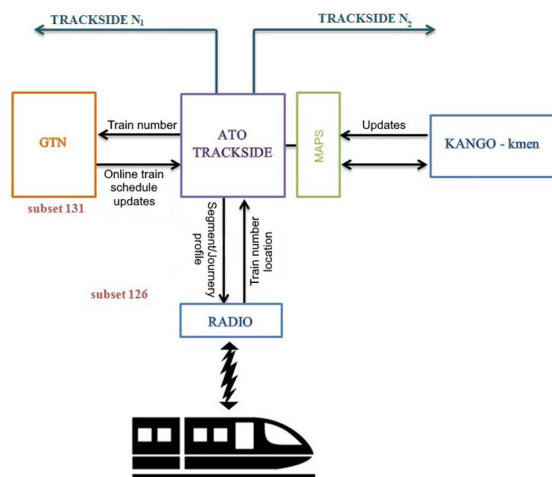


Figure 1. General ATO architecture (AŽD Praha)

Basic ATO structure defined by AŽD Praha can be found in Figure 1. This proposed structure is based on X2Rail-1 [3] project supported by EU Horizon 2020.

In addition to the ATO, various innovative methods of automatic traffic management can be encountered in the railway environment. Some of them focuses on improvements through the convergence of mainline and urban train control systems [4]. Some of outcomes of those are also included in ATO specification. Other work deals with effective communication with trains using modern mobile networks [5]. Automatic train control system is widely used in Asia for a while. They basically the use same principles as designed ATO system [6][7].

We cooperate with AŽD Praha on this project. They are the main contractor for this project. Our goal was design and implement ATO TRACKSIDE part of system.

II. TRAIN CONTROL

Railways are a complex infrastructure system. A significant part of them was built in the second half of the 19th century. They are being modernized over time. A lot of different technologies have been created and used during this modernization.

Railways are used by multiple companies that demands and use different information systems. In this paper, we will focus on systems used by railway networks in the Czech Republic. The core of traffic control is presented by *TMS*, which cooperates dynamic train data. In our project, we are using *TMS GTN*; a system that is used by SŽDC (Railway infrastructure administration in the Czech Republic). The train also needs data of railway infrastructure. Such data are mostly static and in our project are imported from IS *K-Kmen* [9]. This component is part of railway infrastructure systems called *Kango* [10].

A. GTN

Graphic-technological extension (*GTN*) [8] is a complex application that controls and manages: electronic lock-off device complements, station signal boxes, remotely controlled devices of track sections, control of the transportation process itself, etc. *GTN* also includes optimization of transportation control, solving potential conflicts among trains, and keeping transportation documentation. It monitors the *DOZ* system in real-time, and it collects the relevant data on the current situation at individual stations.

GTN makes indicative changes in the organization of transport – operational planning of train work, and it displays the practical implementation of transport on a track section and in individual transport offices, as a list of *GVD* fulfilled.

It works in online mode, and it uses some information on the transport status directly to create a forecasting transport model for the controlled track section. The current data are updated continuously, and consequently, they allow the evaluation of the actual course of the transport process. The manager may follow the results of the model or correct the forecast model manually.

B. ETCS

The ETCS [11] is designed to unify the various national train security systems (Automatic Train Protection). ETCS adds a new level to security systems using “cab signaling”. It brings all important information and track restriction to train operators. It can automatically check the track rules and does not allow its violations.

The ETCS system has two parts. One part is located in the vehicle and the other part consists of physical devices located on track which train passing over during its journey. Compliance with the principles of safe operation is a base premise for the safe automatic train run. ATO achieve this by using data from ETCS within “Full supervision” mode guarantee safe operation.

III. ATO BASIC ARCHITECTURE

There are two main parts of ATO: **unit located on the train (ATO OB)** and the **trackside part (ATO TS)**. For communication parts, we use a protocol that was specially designed by UNISIG (Union Industry of Signalling) for the needs of the project. ATO TS part consists of two sub-parts: server and client parts.

ATO TS server is responsible for data and command exchange with trains ATO OB units. It also gathers and data from external railway systems (yellow color). In our case, ATO uses data from *K-Kmen* (railway infrastructure data) and *GTN* (railway logistic data). For configuration, a web client on the ATO server is used. Whole architecture is shown in figure 2. This figure shows logical connections between the presented components.

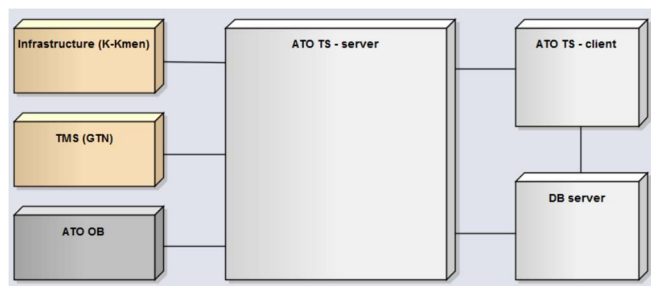


Figure 2. Scheme of ATO system architecture

A. ATO OB

Is represented by a device (computer) located on the train locomotive. After train operator authentication, ATO OB is connected with ATO TS and starts receiving all necessary data about trains, schedules and railway infrastructure. Data are processed and an optimal driving profile is created (which train autonomously follow). If the train uses a track that has no data about, ATO OB must request them from ATO TS. In opposite changes, ATO TS must inform ATO OB (change in train schedule, etc.) If ATO OB doesn't have the necessary data (due to various reasons), ATO is not working

and train operator is prompted to drive train manually. ATO OB is, due to security level, the slave system, train operator can disable or override it, if necessary. In addition, both train operator and ATO are checked by ETCS.

B. ATO TS

One of the main components of the whole ATO system is ATO TS. Its main task is in instruction preparation based on train schedules which are used to train autonomous control.

Initial analysis showed that one ATO TS server will be sufficient pilot functional verification connected to *K-Kmen* and *GTN*.

ATO TS is deployed on a server with known address and individual ATO OBs connect to it. Also, ATO TS is connected with other railway systems, so it keeps actual up-to-date train schedules and actualized data for train management. ATO TS is not monolithic design and its components are described further in section 4.

C. Communication protocol

Each carrier and manager of a railway in the EU can implement components of ATO on its own. Versatility is achieved by necessity to use the ATO UNISIG communication protocol between ATO OB and ATO TS. This protocol is standardized in the Subset - 126.

This protocol contains several versions. If the train wants to use ATO system to manage it, must both ATO OB on the train and ATO TS support the same version of the protocol. The version which will be used in communication is determined in the initial handshake phase.

Communication is realized using the relatively low bandwidth network, so protocol must be minimalistic. The protocol makes possible:

- gain information about train track
- gain and demand information necessary for autonomous train control
- inform ATO TS about current train status

IV. ATO TS ARCHITECTURE

ATO TS is whole build on .net core technology, which makes possible to build our application as multi-platform and modular. During the analysis we identify some services, which can be implemented separately. Some of those services are critical – they have to be functional all the time. Others are needed only for operators when they need to configure the system (prepare data).

ATO was designed to be maintenance-free, so it can operate autonomously by using data from other systems. Nevertheless, there was a request (by train control) to have some possibility to view or edit those automatically prepared data manually. So ATO also contains some services to maintain train data manually.

ATO TS can be divided:

- ATO TS OS - operational services
- ATO TS SS - support services

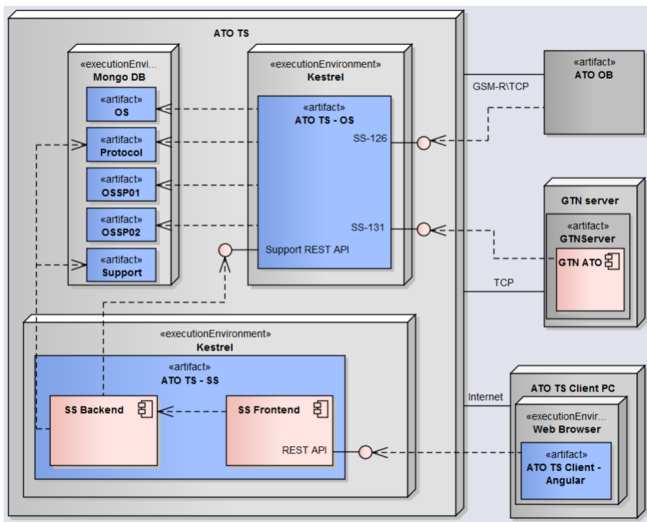


Figure 3. Architecture of ATO TS

A. ATO TS – OS

ATO TS OS is server part written in ASP.net Core which hosting multiple service such as:

- Trains mediator
- ATO OB endpoint
- GTN connector
- Support REST API

This component contains all mission-critical component that has to operate nonstop. Some of those components are stateless (to be quickly restarted). Others, which holds some data must have backup. However, in this pilot project, redundancy was optional, so described architecture doesn't contain redundant components.

Trains mediator is service for train management inside the ATO system and generates data needed for train management. This service communicates with other systems using interfaces.

ATO OB endpoint is a train connection manager. Internally ATO TS uses its own data format and conversion is needed for communication with ATO OB. This component contains a module which is used to code/decode UNISIG protocol messages using the needed version. This is a key part because the new version of UNISIG protocol is released regularly.

GTN connector with the interface defined in subset 131 is specific for our solution. We have to design this protocol by us because the definition of subset 131 is still not final and is currently being discussed on UNISIG. This connector serves for obtaining dynamic data about train schedules. Service implements HTTP endpoint which serves as a connection for GTN. After connection is established ATO can start received paired train data and their schedules.

Last of services is **Support Rest API** which provides data for monitoring and configuration. In its possibilities is to load actual traffic data, making and reload backups and monitoring of system availability.

As DB server we used NoSQL document DBS - *MongoDB*. The whole server consists of five spared databases. First is the database "OS" used for storing runtime data in case of drop-out. It contains a list of trains and data needed for their management. The second database contains protocol. This protocol stores data used for logging purposes. The third database contains data for supporting services ("Support" database). The remaining databases contain data about railway infrastructure. In every moment only one of these databases contains up-to-date data and second future-up-to-date data. The reason for this solution is in the time-consuming process of data preparation. New data are added to inactive DB and when they become valid, the server just switches this DBs.

B. Configuration and monitoring

Configuration and monitoring are not considered mission-critical components. Those services were implemented as separated application. ATO TS - SS is a custom web application created in Angular.js/ASP.net core application. This application provides management of users, management of railway infrastructure and deploys data to production, settings management and monitoring of the system.

Monitoring can provide data about whole and detail ATO OB and ATO TS communication.

C. Deployment

The user architecture of ATO is components oriented towards service architecture. This allows us to use Docker containers to do the deployment. Each of the described services has its own container (figure 4):

- ATO OS container
- ATO SS container
- DB container

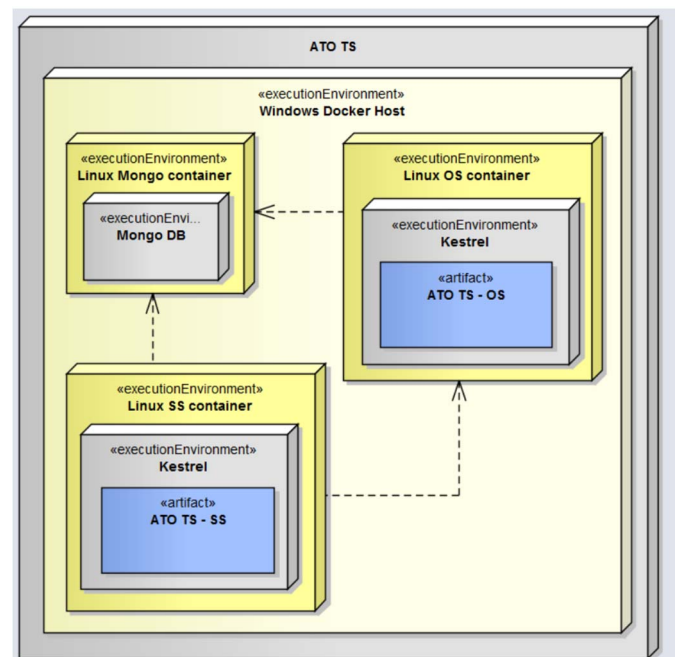


Figure 4. Deployment architecture

Using docker in our solution provides some advantages: no need to specify needed hardware to detail and deployment environment specifications for each component.

In fact, there was only one request that docker host is present. Also, configuration and build of the application is very simplified by using docker-compose scripts, especially for final operator (configuration and environment setup are no needed in such depth).

V. ATO SYSTEM TESTING

Our solution is heavily tested in all its functions: communication between internal and external components, *ATO TS* and *ATO OB* and verification if messages meet *UNISIG* protocol. The testing process has these phases:

1. *ATO* server testing - continual development testing. Verifies: basic function of all *ATO* parts, two-way communication between *ATO TS* and *GTN*, *ATO TS* and *ATO OB*. We create *ATO OB* simulator for early testing needs.
2. *ATO OB* testing: AŽD Praha tested their implementation of *ATO OB* to meet specification and control of locomotive.
3. Laboratory tests - this phase test communication between *ATO TS* and *ATO OB* in test PC in AŽD Praha using real network connection between those systems.
4. Real-world track test - test on real test track Čížkovice - Obrnice in Czech Republic. The test will be done by test scenarios prepared by Shift2Rail.

The first three phases of testing are now completed. Laboratory testing confirmed the functionality of the designed architecture. Tests on the real track are currently under preparation.

VI. CONCLUSION

The main goal of the *ATO* project was implementation of the European train management system on real-world test track Čížkovice-Obrnice in the Czech Republic. Such systems are very beneficial due to their energy-saving potential [12].

An early test was designed, implemented and verified only with systems, allowing operation on a defined single track. Results are showing that a designed solution is

functional. On current state communication protocol is tested for issues with data error or communication errors.

Phase 4 - testing on real-world train track will be done at the end of year the 2019 on test track Čížkovice - Obrnice in the Czech Republic.

After a successful test on the test track project will continue. The biggest challenge of this project will be an expansion of this system to larger areas.

REFERENCES

- [1] Costa, Raúl, and Manuel Villalba. "The Application of the Upcoming Standard on *ATO* over *ETCS*." 2018, www.tecnica-vialibre.es/documentos/Articulos/360AV05_3.6.CostaVillalba.pdf.
- [2] Bienfait, Benoit, Patrick Zoetardt, and B. Barnard. "Automatic train operation: the mandatory improvement for *ETCS* applications." *Hentet April 25* (2012): 2016.
- [3] X2RAIL-1 D3.1 - User & System Requirements (Telecommunications) - available for download on http://projects.shift2rail.org/s2r_ip2_n.aspx?p=X2RAIL-1
- [4] Gurnik, Peter. "Next generation train control (*NGTC*): More effective railways through the convergence of main-line and urban train control systems." *Transportation Research Procedia* 14 (2016): 1855-1864.
- [5] Kim, Juyeop, et al. "Automatic train control over *LTE*: Design and performance evaluation." *IEEE Communications magazine* 53.10 (2015): 102-109.
- [6] Dong, Hairong, et al. "Automatic train control system development and simulation for high-speed railways." *IEEE circuits and systems magazine* 10.2 (2010): 6-18.
- [7] Hengyu, Luo, and Xu Hongze. "An integrated intelligent control algorithm for high-speed train *ATO* systems based on running conditions." *2012 Third International Conference on Digital Manufacturing & Automation*. IEEE, 2012.
- [8] Krsak, Emil, Hynek Bachraty and Vlastimil Polach. "GTN - information system supporting the dispatcher and remote tracks control" In: *Communications: scientific letters of the University of Žilina*. - ISSN 1335-4205. - Vol. 12, No. 3A (2010), s. 65-73.
- [9] Šotek, Karel, Hynek Bachraty, Viliam Tavač and Marek Tavač. "Information system *KANGO 2015*" In *23rd International Symposium EURO-ŽEL 2015: Symposium Proceedings*. Žilina: Žilinská univerzita, 2015. s. 170-179. ISBN 978-80-263-0936-9.
- [10] Šotek, Karel, Emil Kršák, Hynek Bachratý, Viliam Tavač. "New Trends of *ZONA IS* in environment of Slovak railways" *EURO - Žel 2011, 19. medzinárodné sympóziium, 8.-9. jún 2011, Žilina, Slovenská republika, Žilinská univerzita, 2011*. - ISBN 978-80-263-0003-8.
- [11] Abed, Sajed K. "European rail traffic management system-an overview." *2010 1st International Conference on Energy, Power and Control (EPC-IQ)*. IEEE, 2010.
- [12] Caramia, P., et al. "Automatic train operation systems: A survey on algorithm and performance index." *2017 AET International Annual Conference*. IEEE, 2017.

Adoption of cloud and containerization technologies in Mongolia

Baasanjargal Erdenebat
3in Research Group (Martonvásár)
Eötvös Loránd University
Budapest, Hungary
ORCID: 0000-0003-0471-7183

Tamás Kozsik
3in Research Group (Martonvásár)
Eötvös Loránd University
Budapest, Hungary
ORCID: 0000-0003-4484-9172

Abstract—Over the last few years, cloud-based virtualization and containerization are getting momentum day by day. Many countries have been using these novel technologies because of various benefits, such as low overhead, scalability, portability, stability, and high performance. The largest companies have already replaced the traditional architecture with cloud and containerization technologies for their daily operations. However, are developing countries able to keep abreast with this new technological development? How can they adopt these advanced technologies and utilize them for their own benefit? In order to get an answer to these questions, we took a survey from 33 organizations in Mongolia. These organizations are the country's largest representatives of the public and private sectors. The aim of this survey is to evaluate the current state of the system architecture of Mongolian organizations. Moreover, we investigate their readiness to transition to cloud and container technology. Based on the survey result, we identify challenges and make suggestions for improvements.

Index Terms—cloud, containerization, survey

I. Introduction

In the current years, innovation in virtualization technology and new advanced technologies such as cloud virtualization and containerization becomes stronger day by day. Currently, many organizations in many countries use new lightweight technologies because of several benefits such as low overhead costs, scalability, portability, stability, and high performance. Containers are the new revolution in the cloud computing world. They are more lightweight than virtual machines (VMs), and can radically decrease both the startup time of instances and the processing and storage overhead with respect to traditional VMs [1]. Containerization allows lightweight virtualization through the bespoke construction of containers as application packages from individual images that consume less resources and time [2].

As far as Mongolia is concerned, Internet was officially introduced in 1996, and although the history of Information Technology in Mongolia might seem to span for a long time, the actual push that mattered most was between 2000 and 2010, as hardware infrastructure development was the focus during these times. For example, fiber optic cable coverage,

mobile network expansion, the introduction of 3G & 4G technologies and overall use of mobile phones have expanded aggressively, so we can say that the expanse of IT and technical infrastructure was solidified between 2000 & 2010. During this time, and after 2010, the use of various software applications expanded as well.

One can observe that consumers are the main driving point in Mongolia for technological advancement. An example of this is the use of smartphones. In 2013, there were a total of 1,135,464 registered smartphone users, but by 2018, the numbers have shot up to a total of 3,302,000 registered users in gross numbers. As for Internet users, the total registered number was 841,000 users in 2013, but by 2018 the total gross number have become 2,910,000 users [3]. This is a very large number in comparison to the total population of Mongolia which is roughly 3.2 million people [4]. As a result of the growth of Internet users, most organizations need to digitize their business operations. But, compared to cloud-based technology, Mongolians lean more towards smartphones and social media. This can be viewed as a good thing from a consumer standpoint, but note that Mongolians do not have national social media platforms, and the lack of official international technology organizations restricts the advantages from a business standpoint.

According to forecasts, in 2020 around 50% of Mongolian businesses will conduct their operations digitally, and there exists a statistical hypothesis that in 2025, around 80% of Mongolian businesses will be doing so. Also, the Mongolian government has declared many new projects and developed strategies under the influence of the “Digital Government” initiative [5]. The said initiative has helped to improve and expand the ICT field in Mongolia such as switching public service providers to paperless and eco-friendly technologies. Many projects have already been finished and implemented successfully, and there are many more projects underway. The key performance indicator for these technologically-driven digital businesses would be the financial benefits and advantages they can get by using new technologies such as cloud and containerization. Are Mongolian businesses able to adopt and customize these technologies according to their businesses and consumers’ needs and demands? What are the problems they are facing at, and how can we assess them and find solutions?

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

We will be delving deeper into these questions in the following sections. The contribution of this paper is the analysis of the Mongolian technological advances in computing, with a focus on the adoption of cloud- and container-based solutions.

II. Related Work

In 2018, Asia Cloud Computing Association measured cloud readiness index (CRI) among 14 economies across Asia Pacific (APAC) on ten parameters to indicate how prepared they are in adopting cloud computing. The CRI of most APAC economies are as good as that of other eight non-APAC economies. CRI 2018 shows that Singapore tops the APAC region, closely followed by Hong Kong and New Zealand. It also reveals that on average, cloud infrastructure and regulation are the region's strong points, while cloud governance and security weigh it down [6]. India and China have both made significant progress in driving the region's adoption of cloud-driven technologies, playing central roles in the region's technological development. But Regulators still play a pivotal role in facilitating – or inhibiting – the growth of cloud services. In Asia, some regulators are proactively clarifying outsourcing rules and guidelines, while restrictions and obstacles to cloud adoption still exist in others. They singled out Hong Kong, the Philippines, South Korea and Singapore as places that had made progress in eliminating blocks to cloud adoption [7]. However, South Korean regulators still require financial institutions to seek approval before using cloud services, and that personal information is stored within the country [7]. India has recently required all payment service operators to store their payments data within Indian borders to allow unrestricted regulatory access. Indonesia has similar requirements [6] [7].

For the adoption of container technology, many users also choose container-as-a-service offerings from public cloud providers. The AWS container service (ECS/EKS) had 44 percent adoption in 2019 (flat from 2018). Azure Container Service adoption reached 28 percent (up from 20 percent in 2018), and Google Container Engine grew slightly to reach adoption of 15 percent [8]. In the 2019 Container Adoption Survey conducted by 501 IT leaders, sponsored by Portworx and Aqua Security, a whopping 87 percent of respondents said they are now running container technologies, up from 55 percent in 2017 [9]. In Asian countries, there has been a significant jump in the use of containers for testing, up to 42% from 24% in March 2018 with an additional 27% of respondents citing future plans. As the usage of containers becomes more prevalent across all phases of development, the use of container management tools is growing. There has been a significant jump in the usage of nearly all container management tools [10].

As far as we know, there is no other research related to the current state of system architecture and the adoption of cloud and containerization technologies in Mongolia. In 2000, information technology specialist R. Label interviewed over 50 people working in this sector, and created a report about the “Action Plan of the Mongolian ICT field up to 2010” [11].

At that time, he emphasized that the private sectors plays an important role in the development of ICT in Mongolia, unlike other developing countries. The last research of the IT field was in 2010 on the state of the information technology in Mongolia, which is almost a decade ago. It investigated the analysis of the development state of Information technology in various kinds of sectors such as the digital government's initiative, business, education, and health-care [12]. However, they did not go into deep of the main points from the technical sides such as choice of system architecture, main causes of system failures and their impact etc. It was a more general notice which discussed the strengths and weaknesses of IT usage in those sectors from the business perspective.

III. Survey and Collected Data

We have conducted a survey in 2019 to reveal the current state of leading Mongolian organizations with respect to their use of computing technologies. The survey was based on questionnaires and on face to face interviews about the organizations' business activities, system architecture, and recurrent costs. Altogether, 33 organizations have participated in the survey. 48.5% of the total participants were government organizations such as national and science universities, border customs, trade, taxation, election offices and other public service organizations of the Mongolian government. The other 51.5% were non-government organizations, including telecommunications companies, banks, trade & financial organizations, and IT companies. Most of the participants belong to the largest organizations in Mongolia. We observed the total number of servers and the daily user accesses at the participants' site in order to define the magnitude of their business activities, and the potential of system and infrastructure range. Figure 1 illustrates that 46% had more than 30 servers, and the remaining 54% had 1 to 30 servers. Furthermore, 33% of the total participants have less than 500 user accesses per day, but the majority have more than 1500 user accesses daily. We can assume from these numbers that many of the participating organizations operate on quite a large scale.

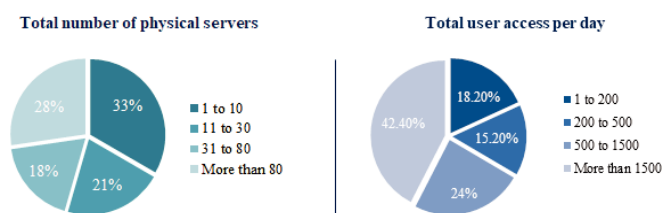


Figure 1. General information of daily user access and physical server

A. Server Downtime and Main Challenges

The survey investigated the frequency of server downtimes per month at the participating organizations, and the key issues that affect the sustainability/availability and reliability of their systems. We have determined that about one quarter of the

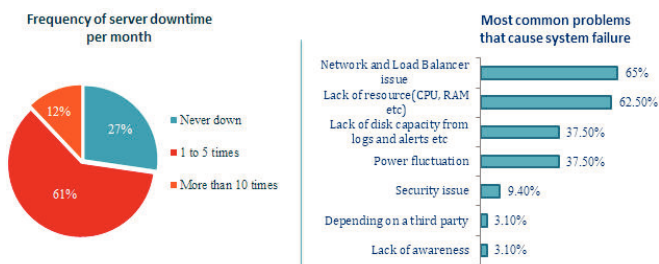


Figure 2. Server downtimes and the main causes

participants can avoid server downtimes, but the majority have problems at providing high quality services to their clients. At the extreme, 12% of the participants experience server downtimes more than 10 times per month, which clearly indicates their inability to provide stable services.

The participants were also given a questionnaire to detail the main problems that cause their systems to fail. Figure 2 illustrates the breakdown of the problems that cause system failures for the participants. As we can see, problems labeled as ‘Network and Load Balancer’ and ‘Lack of resources’ are the most common issues, as over 62% of participants selected them. The next main problems are lack of disk storage, and the fluctuations in electricity, which were selected by 37.5% of the participants. Furthermore, the over-utilization of hardware devices also causes unexpected server downtimes, performance drops, and other issues. Since older devices and equipment tend to negatively impact the reliability and stability of systems and increase risk of system failures, we examined how often the participating organizations replace their server equipments to determine the impact on system availability.

B. Annual IT Budget and Lifespan of Equipment

The lifespan of hardware is also an important part of a system’s performance. On the one hand, out-of-date servers can be more expensive over time because finding replacement parts can be harder, time-consuming and costly, while the server’s performance cannot keep up with the requirements of a developing business. On the other hand, buying new equipment can be high-cost and a huge investment for any company. We collected information on the participants’ capabilities of hardware replacement, and about how this issue impacts their annual IT budget.

As we can see on Figure 3, 52% of the participants answered that they can replace their hardware every 5 to 8 years. We can observe that this is the average term of hardware replacements at Mongolian organizations. Furthermore, we can assume that during the interval of the 5th and 8th year, in which they have not been able to change or upgrade their hardware, the stable operations of their systems and their effectiveness are questionable.

21.2% of the participants answered that they can only replace their hardware within more than 8 years. These participants belong mostly to the government sector. Their annual expenditure is usually planned for maintenance of their

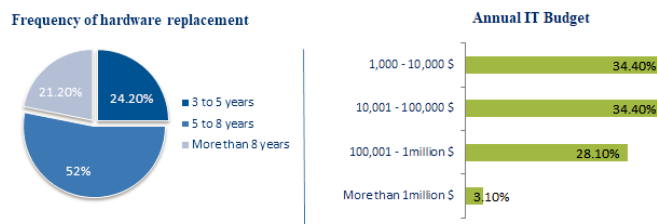


Figure 3. Frequency of hardware replacement and IT budget

existing hardware and component replacements. Furthermore, unfortunately, the government sector’s annual budget for their IT departments is not nearly enough for even maintaining their hardware and replacing components for them. Most of the lower-budget participants (34%) are also in the government sector. They commonly face the main stability problems investigated above during their daily operations, and due to their limited budget, they cannot afford to get professional technical support from third parties either.

Budget problems do not seem this prominent for financial, trade and commercial organizations. They can replace their equipment more frequently (between 3 and 5 years). They realize the benefits and advantages of technology, which ensure their profitability. As we can see, their annual IT budget is also quite high. 32% of the participating organizations spend from 100,000\$ to 1,000,000\$ and more per year on their IT departments. Therefore, their systems are highly stable and are more reliable, and most of them do not have any server downtime.

From the above examination, we have determined that important issues in Mongolia are the high-cost hardware replacements and low budgets for the government sector, but what can be the solution to these – and other related – issues? So we investigated the use of cloud and containerization technologies.

C. Usage of Cloud and container-based technologies

Cloud computing has already gained popularity in various fields, enabling many businesses to access application software over a high-speed internet connection without the need for investing in computer software and hardware [13]. But, in recent years, containerization technology, Cloud containers are a hot topic in the IT world in general, and in security in particular. The world’s top technology companies, including Amazon, Oracle, Microsoft, Google, IBM and Facebook, all use container-based virtualization. From Gmail to YouTube to Search, everything at Google runs in containers [14]. In Figure 4, we can see the main difference between traditional VM architecture and containerized architecture. Containers isolate an application and its dependencies, into a self-contained unit that can run on any platform, anywhere.

This means they will run just as fine on cloud or any standard server so your operations are flexible, reliable and fast. Containers share the machine’s host OS system kernel and therefore do not require an OS per application, which is

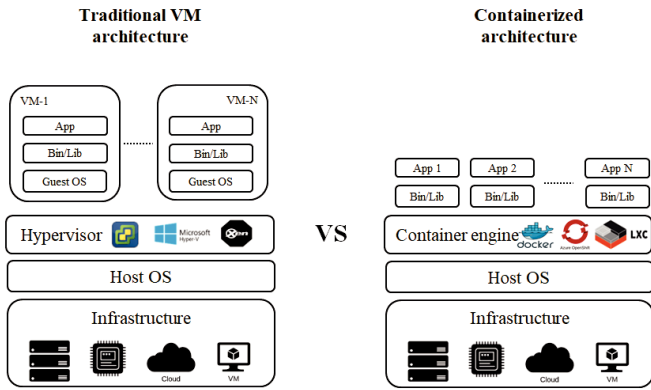


Figure 4. Traditional VM versus containerized architecture

really lightweight and driving higher server efficiencies [15]. Small image sizes make it easier to ship and store when using containers in the cloud. There are several Container run-time environments, including Docker, CoreOS rkt, CRI-O, lxc and others. The set of such environments is steadily growing as the market continues to evolve and diversify. Many organizations are transferring their infrastructure to a containerized one instead of using traditional VMs [8] [16]. Let’s have a brief look at the benefits this new technology gives us.

- 1) Lightweight, Fast & Easy to deploy;
- 2) Independent – no compatibility issues relating to the dependencies / versions / setup, portable anywhere;
- 3) Possibility to orchestrate the cluster containers auto-scale, and self-healing system through container orchestration platforms (Kubernetes, Mesos, Docker Swarm, etc.)
- 4) Reliability, stability, high availability;
- 5) Cost-efficiency.

In 2019, the company Diamanti surveyed more than 500 IT leaders worldwide, and the results reinforce last year’s findings, providing strong evidence that container technology is quickly entering the enterprise mainstream. The use of bare-metal containers is increasing as a way to reduce infrastructure complexity, deliver the full performance and efficiency benefits of the technology, and reduce costs by eliminating licensing (e.g. VMware) [16]. We can see that container based platform adoption rate is continuously growing in the last 2 years from the mentioned survey report. That survey illustrated certain reasons why organizations choose containerization technology. Almost 56 percent who run containers on bare metal do so for

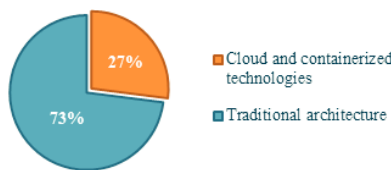


Figure 5. Architecture types used by participants of our survey

higher performance; 36 percent choose bare metal to reduce costs and 29% adopts for less infrastructure [8] [16].

We asked about the usage of cloud and containerization technologies from our survey participants in order to evaluate the stage of the adoption in Mongolia. As seen on Figure 5, 27% of participants answered they use said technologies, but from this, only one organization has fully transferred their whole system to cloud and containerization technologies. What this means is that the positively responding participants (27%) use a combination of traditional VM architecture with “cloud and containerization technologies” such as Docker container and AWS. The remaining 73% responded they use traditional architecture and hypervisor technology such as vSphere, and HyperV.

D. Future plans and Knowledge Level of Technology

In order to forecast participants’ ultimate potential availability of adoption, we investigated the participants’ future plan to transfer their system to the cloud and containerization technologies who using traditional computing architecture. 45.5% of them are planning to use cloud and containerization technologies in the future, and from that number, 71% have already started researching ways to this technology shift. However, the remaining (more than) half of the organizations using traditional architecture are not even planning to use said technology yet (Figure 6).

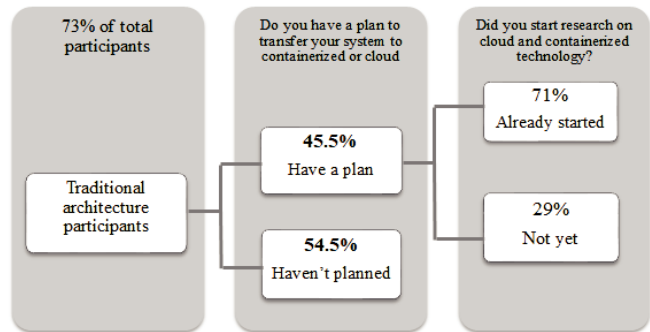


Figure 6. Future plans and research status of cloud and containerization technologies

Also, we tried to measure the knowledge level of the employees of the investigated organizations with respect to cloud and containerization technologies, and their ability to tackle any issues that might come in the future. As shown in Figure 7, about 60% of the employees have low to medium

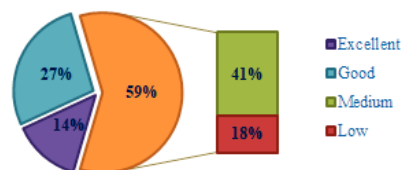


Figure 7. Knowledge level of cloud and containerization technology

level knowledge on the matter, which indicates that our priority should be to focus on training and education of said employees first.

IV. Comparison of costs and architectures

One of the main goals of the current research was to reveal whether a technology shift from traditional architectures to cloud-based and containerized ones would be beneficial to the participating organizations. Therefore we have analyzed incurred costs of both technological approaches. Figure 8 shows the average annual cost of small business entrepreneurs: two servers having average performance. Our calculations include the costs of purchasing physical servers, costs of building a server room according to standards (the above, two costs amortized for 8 years, pessimistically), monthly current expenses and so on. The costs of the same infrastructure when operated in AWS in the cloud have also been estimated. It turns out that it is possible to reduce costs by a factor of 1.3, and that with the AWS approach it is possible to avoid the large initial investment, and to mitigate losses in case of an unsuccessful business.

Type	Cost of buying new server	Preparation cost of new server room	Monthly operation cost	Total cost /amortized - 8 years/
On-premises	~ 7,400\$	~ 1,100\$	110\$ ~ 220\$	19,060 \$ ~ 29,620\$
AWS /Cloud/	-	-	180\$ ~ 230\$	17,280 \$ ~ 22,080 \$ +extra usage charge

Figure 8. Comparison of cost estimates

Figure 9 shows a list of problems that the organizations participating in the survey commonly face. Organizations with traditional computing architecture will face more problems. The ones using cloud technologies need to focus on less issues, since the cloud solves many of the problems. The latter organizations only need to focus on right implementation & customization, IT personnel and training. The cloud data center offers them a more reliable, flexible and cost-efficient system.

As a good practice, we have analyzed the system architecture of one of the largest online stores in Mongolia. Among

Things to consider	Traditional data center	Cloud data center
Network & Security	✓	-
Hardware	✓	-
Resource	✓	-
Maintenance	✓	-
Power outage and fluctuation	✓	-
Accidents (Cyber-attacks, data losses)	✓	-
Implementation & Customization	✓	✓
IT Personnel	✓	✓
Training	✓	✓

✓ You Manage - Vendor Manages

Figure 9. Issues to manage in the IT infrastructure

all similar companies, they were the first to deliver a solution which has a fully cloud-based architecture. The AWS cloud platform has been efficient and fast in their case, with a complete system solution for retaining user search data and for analyzing it. If this system was built by traditional architecture, it would be heavy, costly and would be doubtful whether the system performance was satisfactory at all.

The company used a mirror server in order to maximize local speed, and they rely on Amazon’s featured services lambda, personalize (for AI, machine learning), elastic search, elasticache and so on. The cost estimated for traditional architecture was reduced by 5 times using this technology. The co-founder of this organization said: “It is difficult to build that kind of complete system without cloud technology and AWS new features. These technologies give much more opportunities and save time and labors.”

V. Evaluation of Results

Relying on the survey data, the common issues of system failures and the frequency of downtimes are illustrating a high necessity of expanding, upgrading and re-implementing current systems both at the hardware and the software level. Issues include concerns with network, load balancer, and security. Cloud adoption is one of the best solutions to avoid frequent hardware replacements implied by changing business requirements. Power outage and power fluctuations are rather common in Mongolia, and they are the ultimate cause of equipment corruption and malfunctions.

Organizations should develop optimal solutions for their system based on their particular requirements. For instance, banks and some public sector organizations do not store their data in a public cloud according to the data protection regulatory [17]. In this case, container-based technologies and private or hybrid cloud solutions could be the most suitable approaches for their demands.

As far as the government sector is concerned, the proper working and the effectiveness of their systems in the future are doubtful: the survey results indicate that the foundations of infrastructures and system architectures are not up to basic requirements/standard of system such as sustainability and stability. They definitely need to re-consider either upgrading the whole IT infrastructure, or following modern technologies in their software architectures. Since it is advantageous to discard old, heavy-weight and costly architectures for a better one, the organizations would need to focus on effective redesigning of their infrastructure and applications, as well as to prepare realistic cost & risk assessments, and to make sure that they have professionally qualified experts in the area from the very start, before transferring their systems to cloud and containerization technologies.

According to our survey, one of the main reasons of the slow progress in technology shift is that it is very hard to garner support from the executive levels and employers in Mongolian organizations. They may observe a technology shift counter-productive at first glance – just because these organizations have already spent a large amount of money on their existing

hardware and equipment. However, 80% of the participating organizations still require upgrades, and have equipment they need to regularly renew. Most of the investigated organizations in the private sector have significant annual budgets for their IT departments, but switching to the cloud and containerization technology can give them opportunities to improve sustainability: to re-design and re-establish their current system to a more reasonably priced, more suitable, simple and lightweight architecture.

Adopting cloud and containerization technology requires the following 4 phases from the organization.

- 1) Explore – Stage of research of new technologies & environments.
- 2) Try – Transfer non-mission critical systems to a cloud or containerization technology and run tests on the new environment.
- 3) Expand – Transfer as many non-mission critical systems as possible.
- 4) Transform – Transfer the whole system to cloud or containerization technology, if your team is convinced about its feasibility.

The minority of the participating organizations have already used cloud and containerization technology within production through these phases. Some other participants have been trying to change their monolithic applications to microservices – these activities are still ongoing and this is a good starting point for them. But the adoption rate of cloud and containerization technologies are still low in Mongolia, and according to the survey results, the majority of the participating organizations stand on, or before the first phase, “Explore”. The main blocking issue is the lack of information and knowledge, namely, most of them do not have:

- sufficient knowledge of cloud and containerization ownership;
- sufficient knowledge of cloud and containerization usage;
- a clear approach to the transition to cloud and containerization technologies.

Many of the participating organizations need to research this area from the very beginning and they will also need experts and professionals in this field. Furthermore, they could have the potential to solve their most demanding problems, and could operate more efficiently with the use of cloud and container technologies. For instance, it would be possible to reduce system failures, interruptions, and damages caused by electricity fluctuations and power outages, which is one of the common issues in Mongolian organizations. Also, these technologies would be possible to reduce business profit loss arising from the previously mentioned issues. Cost analyses reveal that rentals and other operational costs are quite high (compared to living costs and the average wage) in Mongolia. The use of new technologies has the potential to reduce costs. Moreover, organizations do not need to care about system interruption and operation failure due to security and load balancing etc. issues.

VI. Conclusion and Recommendation

This paper analyzed the current state of choice of system architecture and the adoption of cloud and container technologies in Mongolia. According to the survey results, it is very rare that organizations participating in the survey have transformed their entire system to cloud and container technology. The majority of these organizations use traditional computing architecture, and more than half of them have not got any plan to adopt cloud and container technologies. Around two-thirds of the organizations do not have sufficient knowledge of said technologies either. From all of that, we can come to the conclusion that the adoption of cloud and container technologies is still in an initial stage in Mongolia.

As we have discussed in Section IV, the adoption of cloud and container technologies can bring a good impact on the overall cost and reliability of the system, especially for SMEs. It has the potential to solve the most common issues of the investigated organizations: hardware replacement, security, downtimes, high investment etc. Relying on the survey result, our primary advice to organizations is to seek for advanced trainings and seminars on cloud and container technologies for improving the knowledge and proficiency of employees. Moreover, we also recommend to carry out adequate research on the re-design of their software architectures, and to introduce the novel technologies gradually.

References

- [1] A. Tosatto, P. Ruiu, and A. Attanasio, “Container-based orchestration in cloud: State of the art and challenges,” in *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, 2015, pp. 70–75.
- [2] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, “Cloud container technologies: A state-of-the-art review,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 677–692, 2019.
- [3] Communications Regulatory Commission of Mongolia, “Main Indicators of ICT Sector 2013-2018 years,” [Online].
- [4] Mongolian Statistical Information Service, [Online].
- [5] Digital Government initiative, [Online].
- [6] Asia Cloud Computing Association, “Cloud readiness index – 2018 report,” [Online].
- [7] —, “Regulatory landscape impacting the use of cloud by financial services institutions in Asia – 2018,” [Online].
- [8] Flexera, “RightScale 2019 State of the Cloud Report,” [Online].
- [9] Portworks and Aqua Security, “2019 Container Adoption Survey,” [Online].
- [10] Cloud Native Computing Foundation, “CNCF Survey: Cloud Usage in Asia,” [Online].
- [11] П. Лабэл, “МХТ-ийн 2010 он хүртэлх үзэл баримтлалыг хэрэгжүүлэх үйл ажилгааны төлөвлөгөө,” (in Mongolian).
- [12] Г. Уртнасан and Б. Цэрэнпунцаг, “Мэдээллийн технологийн орчин үеийн асуудлууд,” (in Mongolian).
- [13] R. Sharma, “The impact of virtualization in cloud computing,” *International Journal of Recent Development in Engineering and Technology*, vol. 3, no. 1, pp. 197–202, 2014.
- [14] Google, “Containers at Google,” [Online].
- [15] Docker, “What is a Container,” [Online].
- [16] Diamanti, “2019 Container Adoption Benchmark Survey,” [Online].
- [17] “Mongolian data protection regulatory,” [Online].

Symbolic Execution with Finite State Automata

Endre Fülöp

ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group
Martonvásár, Hungary
 gamesh411@gmail.com

Norbert Pataki

ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group
Martonvásár, Hungary
 patakino@elte.hu

Abstract—Static analysis is an essential way to find code smells and bugs because it checks the source code without execution. Moreover, static analysis can help in software engineering comprehensively, since static analysis can be used for the validation of code style, evaluate software complexity and execute code refactorings, as well. Symbolic execution is a static analysis method where the variables are interpreted with symbolic values.

Clang Static Analyzer is a powerful symbolic execution engine based on the Clang compiler infrastructure that can be used with C, C++ and Objective-C. Validation of resources’ usage (e.g. files, memory) requires finite state automata (FSA) for modeling the state of resource (e.g. locked or acquired resource). In this paper, we argue for an approach in which automata are in-use during the symbolic execution. In this approach, a generic automaton is used. The generic automaton can be customized for different resources. We present our domain-specific language to define automata. Our tool parses the automaton and generates checker for the symbolic execution engine. We present some generated checkers, as well.

Index Terms—static analysis, Clang, finite state automata, domain-specific language

I. INTRODUCTION

Resource problems (e.g. memory leak) may occur in C and C++ programs because the their validation is not part of the usual compilation process [1]. This kind of problems typically is not covered by unit tests. Special runtime running environment should detect these problem, such as Valgrind [2]. Static analysis is by definition the act of uncovering properites and reasoning about software without observing its runtime behaviour, restricting the scope of tools to those which operate on the source representation, the code written in a single or multiple programming languages. While most static analysis methods are designed to detect anomalies (called bugs) in software code, the methods they employ are varied [3]. One of the main differences is the level of abstraction at which the code is represented [4]. Because static analysis is closely related the compilation of code, the formats used to represent the different abstractions are not unique to the analysis process, but can be found in the compilation pipeline as well [5].

In this paper, we argue for a generic, resource-oriented static analysis approach. One can define special resource-oriented checker with our domain-specific language (DSL) and our

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

tool generate a static analysis method to detect the problems regarding the proposed resource.

The rest of this paper is organized as follows. First, we detail the different levels of static analysis in section II. In section III, we present some typical resource-oriented problems. We show how finite state automata (FSA) help detecting resource problem in section IV. In section V and VI, we define our domain-specific language regarding its syntactic and semantic elements. Integration of our approach with the Clang infrastructure can be seen in section VII. The future work is detailed in section VIII. Finally, this paper concludes in section IX.

II. STATIC ANALYSIS METHODS

A. Textual representation

Some analysis techniques can be run on the source code text that can be seen in Listing 1. This representation is natural for the developer to read, and can be used for review purposes, the code style and the applicability of certain implementation decisions can be analyzed. The main limitation, however is that for tools performing automated analysis the stream of tokens, this representation can prove to be insufficient source of information. The tokenized source code does not reveal the structure of the software, which is normally added by the parser. Solutions formulated at this abstraction level are also sensitive to reformatting, and refactoring transformations applied to the source code. As a result of these properties, it is infeasible to implement checks for complex properties in a stable, reusable fashion.

```
#define Z 0
#define DIV(a, b) ((a)/(b))

int main()
{
  int a = 1 / 0; // found trivially
  int b = 1 / Z; // needs preprocessor
                information
  int c = DIV(1, 0); // same as above

  // not detected by naive implementations
  int d = 1 / (1 - 1)
  int e = DIV(1, 1) / (DIV(1, 1) - DIV(1, 1))
}
```

Listing 1. The capabilities and shortcomings of textual representation

B. Abstract Syntax Tree

Compared to the textual representation, more elaborate of analysis problems can be solved if the structure—dictated by the formal grammar of the programming language—is provided. In addition to revealing structure, type information also becomes accessible. The parser is the most common source of the abstract syntax tree (AST), as it is also used for code generation. This also means parsing of the source code is a prerequisite for static analysis using the AST [6]. Another drawback of this representation is that the dynamic properties of the software are still hidden, the possible execution paths cannot be taken into account when defining the detection criteria of an analysis solution. This level of abstraction can be utilized to detect errors similar to the problem in Listing 2. In the Clang infrastructure, there is an existing embedded domain-specific language for matching the AST, called the ASTMatchers library (Listing 3), which provides a declarative interface for selecting, traversing, nodes of the AST, and also allows one to bind them to symbols, which can later be used as input for other matchers, or for generating diagnostics [7].

```
// Clang Tidy check: cppcoreguidelines-slicing
struct B
{
    int a;
    virtual int f();
};

struct D : B
{
    int b;
    int f() override;
};

void use(B b) // Missing reference intended?
{
    b.f(); // Calls B::f.
}
//...
D d;
use(d); // Slice.
```

Listing 2. A check based on the syntactic information provided by the AST

```
// Clang ASTMatchers example
// This binds the CXXRecordDecl with name "::
// MyClass"
// to "myClassDecl".

recordDecl(hasName("::MyClass")).bind("
    myClassDecl")
```

Listing 3. Declarative matching of AST

C. Program flow

The AST alone is generally not sufficient to effectively detect problems like in Listing 4. If a static analysis method produces many false positives, it becomes harder to find real programming errors in code, which in the long run reduces the effectivity of the method. In order to reason the program flow, various data-flow analysis methods exist, which can also be found inside the compilation pipeline—often inside the

code optimizer—as well, but using their results for analysis purposes can be challenging.

```
enum Choice { YES, NO };

int f(Choice c)
{
    if (c == YES) { return 1; }
    else { return 2; }
}

int main()
{
    int a = f(getInputChoice());

    // safe to assume that a cannot be greater
    // than 2
    // no matter what the initial choice was
    if (a > 2)
    {
        // unreachable
        return 1 / 0;
    }
}
```

Listing 4. False positives can occur due to insufficient information

D. Program path

Clang Static Analyzer uses an even more detailed model to aid the formulation of static analysis solutions. The program path analysis explores all possible execution paths of the program (Listing 5), and—in case of the Clang Static Analyzer—uses symbolic execution in order to reason about the values along a possible path [8]. The drawback is that the number of execution paths to explore is exponential in the number of branches. This means that practical program path analysis solutions involve some heuristics to select a subset of possible paths to be analyzed [9].

```
int main()
{
    // the return value of 'getInputChoice()'
    // is assumed to be one of the 2 enum values
    // or unknown
    int a = f(getInputChoice());
}
```

Listing 5. Values are modelled along execution paths

III. RESOURCE-LIKE PROBLEMS

Resource-like problems involve a service of the running environment, which can be used by the program via calls to an API [10]. For instance, Linux systems have standard way of interfacing file system. This API describes the correct usage of descriptors and function calls that can be seen in Fig. 1. Generally, these APIs also dictate some rules as for the usage of the individual parts. These can include restrictions on the multiplicity of service objects, the immediate order of triggering events concerning these objects, or eventual state of the service infrastructure. If these rules are not followed by the developer, usually some negative consequences ensue, like resource-overuse and race conditions [11]. The summed

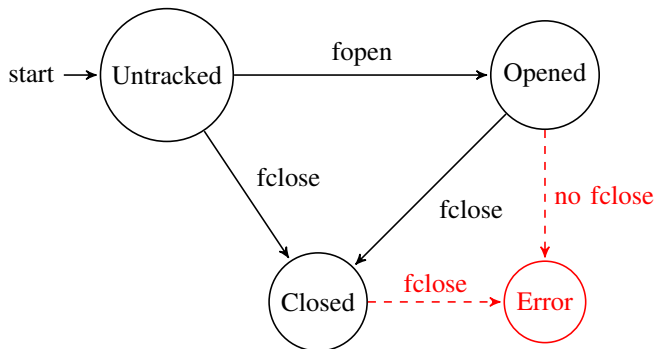


Fig. 1. File handling API model

complexity of programming language constructions, the problem domain logic, and the resource handling may cause errors which are not easily detectable or reproducible [12]. Static analysis can be used to detect violations of such contracts [13].

State machine representation of resource-like problems has another benefit. The level of abstraction can be raised, and many users of static analysis tools can better formulate the error conditions by specifying the abstract machine and its working logic, than starting with the highly detailed view. This applies to solutions implemented in functional, as well as in procedural frameworks, as the level of technical detail to implement a checker is high.

In order to alleviate the implementation burden, several tools and libraries have already been implemented by the Clang Static Analyzer community. Most notably, there is the ASTMatchers library which generalizes the selection of interesting nodes from the translation unit, by giving a declarative embedded domain specific language. Another tool is under development, which would allow the implementer to specify interesting pattern inside the Control Flow Graph (CFG) of the examined software. This would allow for matching against more dynamic aspects of the program execution, however the authors consider this solution more in-line with the aforementioned ASTMatchers library, providing an extra level of precision without the burden of verbosity. It is important to note, that both tools mentioned come with a cost. ASTMatchers uses dynamic memory allocation with reference counting, thus if misused, the memory handling cost could prove to be significant compared to the execution of matching relevant nodes.

IV. GENERALIZING ERROR DETECTION WITH FSA

As discussed above, the finite state automata model is used in the solution proposed by the authors to give a solution for describing error conditions in a program. This solution is in a form of an external DSL. The language lets its user specify a labelled state machine, that's transitions are governed by the relevant events occurring during the symbolic execution paths. The exploration of said paths, and the generation of base events are supposed to be done by the framework. The authors implemented a concrete solution using Clang Static

Analyzer as the backend which provides these facilities. This separation of concerns would allow to use this DSL to be used with other static analysis frameworks, and even other languages (besides C and C++, relevant to the example).

The main idea behind the error detection solution is to separate the matching or filtering into three distinct levels. These three are: *syntax-level*, *value-level*, and *model-level* matching of the relevant events. Also the priority of handling these levels is important. The traditional way of implementing checkers in case of Clang Static Analyzer is to implement call-back functions, which handle events, that fire when symbolic execution reaches certain syntactic categories. This would make the traditional approach *syntax-first*. This means it is more fitting for describing problems that are syntax-oriented, and would have to be more verbose in order to implement state-related checks. The proposed solution is *model-first* in contrast.

V. CHECKERLANG DSL

The implementer could define the state machine (modelling Fig. 1 example) by explicitly specifying the transitions as seen in listing 6. This example excludes the error states, and transitions. The example defines a checker named `stream_checker`, which has a state machine (this is implicit, currently a single state machine kind can be defined for a checker, out of which multiple instances can be instantiated). The machine is defined by a list of transitions. Transitions have a head, a body and a tail part. The head part signifies the starting state, the tail part the destination state. The body part defines the guard conditions of the transaction, as well the extraction of symbolic values, and diagnostics to be emitted. The extraction of symbolic values has a form of traditional assignment expressions, but with a predefined set of extractor functions allowed on the right hand side. The definition of states is implicit, the set of defined states is said to be union of states appearing in the head and body parts of the transactions. The machine described in the example has three transitions, the first two of which is describing how to construct a machine. The start state of the conceptual machine is not modelled, instead transitions with the keyword *construct* in their head signify how to start tracking a symbolic value with a machine. Every constructor transition must have an extractor expression specifying the ID label. This value is used then later to track the state, and perform the necessary transitions. The body part consists of arbitrarily ordered list of either extractor expressions, predicate expressions or diagnostic expressions. Extractor expressions define the labels of the state machine. These labels are persistent across transitions, so they do not follow lexical scoping. This also means, that label ID is readily available in every non-constructor transition. Predicate expressions govern the execution of the transition, the success of a transaction is dependent on their runtime evaluation. Diagnostic expressions help formulate the output of the analysis.

```

checker stream_checker
{constructor}

```

```

    ID = RESULT_OF(fopen)
{file_opened}

{constructor}
    ID = NTH_PARAM_OF(fclose, 1)
{file_closed}

{file_opened}
    fd = NTH_PARAM_OF(fclose, 1)
    fd == ID
{file_closed}

```

Listing 6. DSL to describe FSA

```

(* main language *)
checker_name = identifier;
checker_specification = "checker",
    checker_name;
head_id = identifier;
transition_head = "{" , head_id, "}";
transition_body = (sym_definition |
    sym_predicate | diag_expr)*;
tail_id = identifier;
transition_tail = "{", tail_id, "}";
transition_specification = transition_head,
    transition_body, transition_tail;
transitions = transition_specification*;
grammar = checker_specification, transitions;

```

Listing 7. EBNF of Checkerlang main language

```

(* sym_definition sublanguage *)
result_of_function_name = identifier;
result_of_expr = "RESULT_OF(",
    result_of_function_name, ")";
nth_param_of_function_name = identifier;
nth_param_of_argument_position = digit;
nth_param_of_expr = "NTH_PARAM_OF(",
    nth_param_of_function_name, ",",
    nth_param_of_argument_position, ")";
sym_expr = result_of_expr | nth_param_of_expr;
sym_definition_lhs = identifier;
sym_definition_rhs = sym_expr;
sym_definition = sym_definition_lhs, "=",
    sym_definition_rhs;

```

Listing 8. EBNF of Checkerlang extractor sublanguage

```

(* sym_predicate sublanguage *)
sym_predicate_lhs = identifier;
sym_predicate_rhs = identifier;
op_kind_eq = "==";
op_kind_neq = "!=";
sym_predicate_op = op_kind_eq | op_kind_neq;
sym_predicate = sym_predicate_lhs,
    sym_predicate_op, sym_predicate_rhs;
}

```

Listing 9. EBNF of Checkerlang predicate sublanguage

```

(* diagnostic sublanguage *)
leak_diag = diag, string;
diag = diag, string;
diag_expr = diag | leak_diag ;
}

```

Listing 10. EBNF of Checkerlang diagnostic sublanguage

VI. SEMANTICS OF CHECKERLANG DSL

The transitions defined by Checkerlang are to be executed on a best-efforts basis. This means that the transition described by the language (Listing 7) should be executed like this: try to execute all statements of the transition body in their original order. If a state extractor statement (Listing 8) is encountered, evaluate the expression on the right hand side, then update value of the label indicated by the left hand side. In case of a symbolic predicate statement, evaluate left and the right hand side in no particular order, then evaluate the whole expression. If the value of the whole symbolic predicate statement (Listing 9) is true, then proceed to the next statement. If the value is false, cancel the transaction. This signifies an aborted transaction, however any labels updated so far are not rolled back to their original state. Besides a symbolic predicate expression evaluating to false, any errors during any evaluation also causes abortion. If the statement list is processed without abortion, the transition is deemed successful, and the machine transitions into the destination state. When a diagnostic expression is met (Listing 10), depending on the type of the diagnostic expression one of the following would happen: immediate diagnostic expression triggers the emission of a bug report immediately. There is no way to cancel an immediate diagnostic. In case of leak diagnostics, the diagnostic message is saved for later use. If the machine goes out scope during symbolic execution (this means that the symbol associated with its ID label is considered dead by the execution framework), this diagnostic is emitted. The emission of such diagnostic is however cancelled if any later transaction is successfully executed. These rules help formulate the checking logic in high abstraction level.

VII. PRACTICAL USAGE WITHIN THE CLANG INFRASTRUCTURE

We have implemented a parser to the Checkerlang DSL. This parser uses the PEG parser combinator header-only library PEGTL [14]. The library written in C++ programming language, and allows the assembly of parser from subparsers. The composition mechanic is implemented by inheriting from class templates, and makes heavy use of template metaprogramming [15]. The generated parser is a top-down recursive descent parser, which does not have a separate scanner. Lexing phase is completely omitted, and built in the parser using whitespace-accepting combinators. As the code example 11 shows, implementing the DSL via parser combinators produces succinct, and readable parser implementation.

```

struct ws : one< ' ', '\t', '\n', '\r' > {};
struct wss : star< ws > {};
struct wsp : plus< ws > {};

struct checker_name : identifier {};
struct checker_specification : seq< S("checker
    "), wsp, checker_name > {};
struct head_id : identifier {};
struct transition_head : seq< one< '{' >, wss,
    head_id, wss, one< '}' >> {};

```



```

struct sym_definition : sym_extractor_lang::
  rules::sym_definition {};
struct sym_predicate_expr : sym_predicate_lang
  ::rules::sym_predicate_expr {};
struct transition_body : list< sor<
  sym_definition, sym_predicate_expr,
  diag_expr>, wsp > {};
struct tail_id : identifier {};
struct transition_tail : seq< one< '{' >, wss,
  tail_id, wss, one< '}' >> {};
struct transition_specification : seq<
  transition_head, wss, transition_body, wss
  , transition_tail > {};
struct transitions : list<
  transition_specification, wsp > {};
struct grammar : seq< wss,
  checker_specification, opt< wsp,
  transitions >, wss > {};

```

Listing 11. Implementating with parser combinators

We have created a project, where this header-only implementation of the parser is used to create a Clang plugin. The plugin itself is a dynamically-linkable file, that implements a Clang Static Analyzer checker. This checker can be parametrized with a filename, which contains the specification of the static analysis problem written in Checkerlang, parses it during execution, and directs the analysis and the emission of diagnostics during runtime. This solution enables the rapid development of static analysis solutions, as no recompilation is needed to test independent checks, only parametrizing the Clang invocation differently can result in injecting a brand new checker logic.

VIII. FUTURE WORK

The extractor language has some limitations at the moment. In order to harness the power of tools already developed, namely the ASTMatchers library, the extractor language should be extended. A trivial extension would be to create an expression extractor sublanguage as well, integrate it into the current symbolic extractor language. The expression extractor sublanguage could be the ASTMatchers language itself. Another extension would be to allow multiple kinds of machines to be defined in a checker. The instances of multiple machines could share state, or pass messages between each other to coordinate for better solutions. The semantics of these extensions are to be defined later on, as well as the performance cost of using dynamic linking, and interpreting the the checking logic.

IX. CONCLUSION

Resource problems in C/C++ programs are quite typical. Compilers cannot detect many problems regarding resources, like double free, leakage, etc. In this paper, we present a generic approach that aims at static analyses of resource management. Our approach involves a new domain-specific language, called Checkerlang. Our tool is able to parse Checkerlang source code and it generates Clang Static Analyzer checker from the source code.

REFERENCES

- [1] S. Meyers, “Effective C++”, 3rd Edition, Addison-Wesley, 2005.
- [2] N. Nethercote, J. Seward, “Valgrind : A framework for heavyweight dynamic binary instrumentation,” in Proc. of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’ 07), 2007, pp. 89–100.
- [3] B. Johnson, Y. Song, E. Murphy-Hill, R. Bowdidge, “Why don’t software developers use static analysis tools to find bugs?,” in Proc. of the 2013 International Conference on Software Engineering (ICSE ’13), 2013, pp. 672–681.
- [4] C. King, “Symbolic execution and program testing,” *Commun ACM*, vol. 19, pp. 385–394, 1976.
- [5] N. Nagappan, T. Ball, “Static analysis tools as early indicators of pre-release defect density,” in Proc. of the 27th International Conference on Software Engineering (ICSE ’05), 2005, pp. 580–586.
- [6] B. Babati, G. Horváth, V. Májer, N. Pataki, “Static analysis toolset with Clang,” in Proc. of the 10th International Conference on Applied Informatics, ICAI 2017, 2017, pp. 23–29.
- [7] Clang AST matcher library, <http://clang.llvm.org/docs/LibASTMatchers.html>
- [8] G. Horváth, N. Pataki, “Source language representation of function summaries in static analysis,” in Proc. of the 11th Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOLPS’16), 2016, pp. 6(1)–6(9).
- [9] Cs. Szabó, M. Kotul, R. Petruš: „A closer look at software refactoring using symbolic execution,” in Proc. of the 9th International Conference on Applied Informatics (ICAI 2014) vol. 2, 2014, pp. 309–316.
- [10] S. C. Dewhurst, “C++ gotchas avoiding common problems in coding and design,” 2003, Pearson Education.
- [11] B. Stroustrup, “The C++ programming language,” 4th Edition, Addison-Wesley, 2013.
- [12] D. Papp, N. Pataki, “Bypassing memory leak in modern C++ realm,” *Annales Mathematicae et Informaticae* vol. 48, 2018, pp. 43–50.
- [13] J. Slabý, J. Strejček, M. Trtík, “Checking properties described by state machines: On synergy of instrumentation, slicing, and symbolic execution,” in *Formal Methods for Industrial Critical Systems*, 2012, pp. 207–221.
- [14] Parsing Expression Grammar Template Library, <https://github.com/taocpp/PEGTL>
- [15] Z. Porkoláb, Á. Sinkovics, “Domain-specific language integration with compile-time parser generator library,” in Proc. of the Ninth International Conference on Generative Programming and Component Engineering (GPCE 2010), ACM SIGPLAN Notices vol 46(2), 2010, pp. 137–146.

Cyber Rapid Response Team: An Option within Hybrid Threats

1st Darko Galinec

*Department of Informatics and Computing
Zagreb University of Applied Sciences
Zagreb, Croatia
darko.galinec@tvz.hr*

2nd William Steingartner

*Faculty of Electrical Engineering
and Informatics
Technical University of Košice
Košice, Slovakia
william.steingartner@tuke.sk*

3rd Vinko Zebić

*Croatian Military Academy
"Dr. Franjo Tuđman"
Croatian Armed Forces, Ministry of Defence
Zagreb, Croatia
vinko.zebic@morh.hr*

Abstract—The “rules of war” themselves have changed significantly. Nonmilitary options have come to play a greater role in achieving political and strategic goals and, in some situations, are greatly superior to the power of weapons. The role of mobile joint forces operating in an integrated reconnaissance and information environment is rising through the use of new opportunities now available to control and logistic systems [7]. The European Union (EU) and its Member States (MS) continue to face serious and acute threats, which are increasingly taking non-conventional forms, such as radicalization leading to terrorist attacks, chemical attacks, cyber-attacks or disinformation campaigns. All these actions have one thing in common - they seek to destabilize and endanger society and undermine core values. In connection with the foregoing authors created and presented the novel model of hybrid threats. Furthermore, within the same model authors investigate actions for cybersecurity and cyber defence in conditions of increasing challenge of cyber-attacks and the limited capabilities to respond to this threat describing the process of creation and performance of EU Cyber Rapid Response Teams (CRRTs) and Mutual Assistance in Cyber Security, introducing novel approach to cybersecurity and cyber defence at the EU level.

Index Terms—cyber-attack, cyber rapid response teams, cyber threats, hybrid threats

I. INTRODUCTION

Hybrid threats combine conventional and unconventional, military and non-military activities that can be used in a coordinated manner by state or non-state actors to achieve specific political objectives. Hybrid campaigns are multidimensional, combining coercive and subversive measures, using both conventional and unconventional tools and tactics. They are designed to be difficult to detect or attribute. These threats target critical vulnerabilities and seek to create confusion to hinder swift and effective decision-making. Hybrid threats can range from cyber-attacks on critical information systems, through the disruption of critical services such as energy supplies or financial services, to the undermining of public trust in government institutions or the deepening of social divisions. Chemical, Biological, Radiological and Nuclear (CBRN) threats delivered by non-conventional means fall

This publication was produced thanks to the support under the Operational Program Research and Innovation for the project: Intelligent Data Infrastructures for the Digital Society, co-financed by the European Regional Development Fund.

within a category of their own because of the potential scale of the damage they can cause. As attribution is difficult, these challenges require specific and coordinated measures to counter; for example, detection of the transfer of dangerous chemicals, reducing access to them, or decontamination. The aim of this paper is to construct the novel model of hybrid threats as well as to investigate actions for cybersecurity and cyber defence in conditions of increasing challenge of cyber-attacks and the limited capabilities to respond to this threat. Secondly, the aim is to describe the process of creation and performance of EU Cyber Rapid Response Teams and Mutual Assistance in Cyber Security, introducing a novel approach to cybersecurity and cyber defence at the EU level putting it into the context of the aforementioned developed model.

II. BASIC NOTIONS ON HYBRID THREATS AND CYBER RAPID RESPONSE TEAMS

Awareness, resilience and response are at the heart of EU action to counter hybrid threats. EU is improving the capacity to detect and understand malicious activities at an early stage. At the same time, the EU enhances the resilience of critical infrastructure, societies and institutions. This is fundamental to improving the ability to withstand and recover from attacks. Countering hybrid threats requires action mainly from the Member States, as well as closer cooperation between the EU, the Member States, partner countries and NATO.

A. Hybrid Threats

New information technologies have reduced appreciably the distance – physical, temporal, and informational – between the troops and their superiors. Remote engagement of the enemy “at arm’s length” is turning into the principal tactic to achieve the goals of combat action or operation. Adversary targets are now attacked at any point of enemy territory. Differences between strategic, operational, and tactical actions, and between offense and defense are leveling off. High-precision weapons are used on a growing scale. Weapons based on new physical principles and robot-controlled systems are going into service in large quantities [7]. Rapid development of information technologies in the late 20th and early 21st centuries, and widespread use of information in society and the



Figure 1. Hybrid Warfare

leading countries' armed forces have changed significantly the nature, methods, and techniques used by state and government political and economic agencies, affected social relationships and the nature, methods, and techniques of military operations, and created new information threats and challenges [8].

Hybrid threats [13]:

- are coordinated and synchronized,
- deliberately target democratic states' and institutions' systemic vulnerabilities,
- use a wide range of means,
- exploit the threshold of detection and attribution as well as the border between war and peace,
- are aimed to influence different forms of decision making at the local (regional), state, or institutional level.

Hybrid Warfare is a combination of multiple conventional and unconventional tools of warfare (Figure 1) [13].

B. Cyber Rapid Response Team

In the run-up to his special operation, the attacker will presumably make wide use of nonmilitary (indirect) moves and techniques, including targeted cyber-attacks against the communications systems of the enemy's control bodies at all levels. decisive battles in new-generation wars will rage in the information environment, in which the attacker's computer operator manipulating the "intelligent machines" at a distance will be the key figure in the battle-space.

Encrypted data flowing in public communication channels will be among the coveted targets for cyber-attacks.

A quantum computer [18] may turn into a tool of destruction and a 21st-century bomb for cyber-attacks to succeed. It will easily crack all codes and gain free, and virtually instant, access to all networks supporting the operation and security of government and military control agencies [14].

Cyber defence focuses on preventing, detecting and providing timely responses to attacks or threats so that no infrastructure or information is tampered with. With the growth in volume as well as the complexity of cyber-attacks, cyber

defence is essential for most entities in order to protect sensitive information as well as to safeguard assets [16].

Cyber defence provides the much-needed assurance to run the processes and activities, free from worries about threats. It helps in enhancing the security strategy utilizations and resources in the most effective fashion. The cyber defence also helps in improving the effectiveness of security resources and security expenses, especially in critical locations [6].

Understanding an increasing challenge of cyber defence and the limited capabilities to respond to this threat, Lithuania proposed to the EU Council on Defence a project on Cyber Rapid Response Teams and Mutual Assistance in Cyber Security which aims not only to strengthen own security but as well to increase cyber defence capabilities on the European level.

By the project, it's intended to create multinational rapid response cyber teams composed of participating countries' cyber defence experts. The value-added of the project is that different from many other existing multinational initiatives in cyber defence which concentrate on the exchange of information this project will include sharing of the human resources.

The project will cover research on various legal procedures in the domain of cyber security in the EU, organization of tabletop exercises (cyber crisis simulation exercises) and development of cyber defence tools. Eight EU countries have already joined the project (Croatia, Estonia, Finland, Italy, Lithuania, Netherlands, Poland, Romania), six states are observing the project (Belgium, France, Germany, Greece, Slovenia and Spain).

III. HYBRID THREATS MODEL

Realities of our day dictate an urgent need to prepare for possible new-generation wars without further delay. It is a first priority for the country because its defense and economic potential has waned significantly over the past two or three decades.

Every effort must be made to repair deteriorating relations between states before a new-generation war breaks out, preferably beginning with nonmilitary options, such as a combination of political, economic, scientific, engineering, information, diplomatic, and humanitarian opportunities of a country to be integrated into the world community and versatile relations to be developed with it by confidence-building measures and expansion of its cooperation with other countries, to ease military confrontation, and to put up barriers to power politics [15].

A. Nonmilitary Options

These nonmilitary options will lessen, and ultimately remove, military hazards and threats by peace treaties and other initiatives, reduce the aggressor's choices in his hostility to other countries, give him an unfavorable image, and expose his aggressive plans. Where nonmilitary efforts produce little effect, a country must be ready to use every kind of power containment to persuade the potential aggressor that the costs of his attack will be higher than the expected results.

The enemy may be swayed in his resolve by demonstration of the readiness (in response to a threat of attack) of a defensive force to be deployed to the area of anticipated aggression; a strongly worded statement with a warning of immediate nuclear retaliation against the threat arising to the country's sovereignty and integrity during the war and of unrestrained use of high-precision weapons to destroy the enemy's nuclear power plants, chemical industry plants, and major hydropower projects on the potential aggressor's territory; and preparation and conduct of an information operation expressly to mislead the enemy about readiness to fight off aggression.

We are almost certain that as a new world order is put in place, armed force will, in the short term, continue to play a major role in the efforts economically. Advanced countries and their allies will be making to achieve their political objectives. To avoid making the same historical mistake yet another time, the armed forces must be ready to fight new-generation wars in the medium and long terms and to use indirect, arm's length forms of operations [15].

Information superiority and anticipatory operations will be the main ingredients of success in new-generation wars. The intensity of military operations will peak from the start, with the attacker entertaining the hope of striking a first surprise and most powerful and crippling blow. A country preaching a defensive doctrine may get the short end of the deal in the face of a surprise attack by an aggressor [1].

B. New Generation War

To cope with the growing challenges, which today are manifested as unknown unknowns, systems tend to enable personnel and develop new processes, organization and technology. Technologies are being developed which, unlike traditional approaches [19], have the ability to protect systems from serious threats by learning what is "normal" for the organization and its people and thereby spotting emerging anomalies. Unlike the traditional rules and signature-based approach, the technology can spot threats that could harm the organization and network that the traditional approaches would be unable to detect. It can deal with uncertainty and delivers adaptive protection for organizations from both insider threats and advanced cyber-attacks [5].

A new-generation war will be dominated by information and psychological warfare that will seek to achieve superiority in troops and weapons control and depress the opponent's armed forces personnel and population morally and psychologically. In the ongoing revolution in information technologies, information and psychological warfare will largely lay the groundwork for victory.

Asymmetric actions, too, will be used extensively to level off the enemy's superiority in armed struggle by a combination of political, economic, information, technological, and ecological campaigns in the form of indirect actions and nonmilitary measures (see Figure 2).

In its new technological format, the indirect action strategy will draw on, above all, a great variety of forms and methods of nonmilitary techniques and nonmilitary measures,

including information warfare to neutralize adversary actions without resorting to weapons (through indirect actions), by exercising information superiority, in the first place. Interstate differences preceding new-generation warfare will be resolved by relying on a combination of political, economic, scientific, engineering, religious, cultural, information, and humanitarian capabilities of a country to integrate it into a peaceful environment, develop its diversified relationships that promote trust and cooperation, scale down military confrontation, and put up barriers to power politics.

Cyberspace is the fifth domain of operations, alongside the domains of land, sea, air, and space: the successful implementation of EU missions and operations is increasingly dependent on uninterrupted access to secure cyberspace, and thus requires robust and resilient cyber operational capabilities [2].

Within the non-hierarchical Hybrid Threats Model, military and nonmilitary activities using conventional and unconventional tools and tactics are combined.

Interactions between some individual layers of the model by the means of cybersecurity and cyber defence are being achieved in cyberspace as the common denominator of particular components of the model, enabling multiple threats to be realized through system's (state's) vulnerabilities.

These nonmilitary actions will help lessen and remove military hazards and threats by the opponents entering into peace treaties and taking other amicable steps. Nonmilitary measures serve to reduce the possibility for the aggressor to engage in hostile activities against other countries, give it an unflattering image in public opinion, make sensational denunciations of its aggressive plans, and so on.

Beyond a shadow of a doubt, the aggressive side will be first to use nonmilitary actions and measures as it plans to attack its victim in a new-generation war. With powerful information technologies at its disposal, the aggressor will make an effort to involve all public institutions in the country it intends to attack, primarily the mass media and religious organizations, cultural institutions, nongovernmental organizations, public movements financed from abroad, and scholars engaged in research on foreign grants. All these institutions and individuals may be involved in a distributed attack and strike damaging point blows at the country's social system with the purported aims of promoting democracy and respect for human rights [1].

C. Future EU Efforts

Expanding the capacity to detect hybrid threats and scale up the measures against disinformation campaigns means efforts to counter and respond to hybrid threats have to be underpinned by a capacity to detect malicious activities and their sources, which can come both from within and outside the EU, at an early stage and to understand links between often seemingly unrelated events by connecting the dots.

- Building up the preparedness against CBRN attacks: develop a list of chemical substances posing a particular threat, as a basis for action to reduce their accessibility; set up a dialogue with private actors in the supply chain



Figure 2. Hybrid Threats Model

to work towards restricting the availability of chemicals that can be used as precursors; make a complete overview of threat scenarios and an analysis of existing detection methods to improve the detection of chemical threats.

- Reinforce cybersecurity measures: the EU is aiming to build up capacity through support measures, stronger coordination and new structures to improve counter-measures and accelerate their deployment.
- Reinforce counter-intelligence expertise at EU level: concrete practical measure will be put in place to step up coordination among and between Member States and other international partners, such as NATO.
- Improved capacity to detect hybrid threats: efforts to counter and respond to hybrid threats have to be underpinned by a capacity to detect early malicious hybrid activities and sources. To this end, the EU Hybrid Fusion Cell will be expanded with specialized chemical, biological, radiological and nuclear, counter intelligence as well as cyber analytical components [4].

IV. CASE STUDY

The initiative on creation of joint EU CRRTs and Mutual Assistance in Cyber Security project is among the most advanced projects out of the overall 17 approved in late 2017 under the EU Permanent Structured Cooperation (PESCO) framework.

PESCO is an instrument laid out in the Treaty of Lisbon, for deepening the cooperation in security and defence area for those EU member states that have military capabilities meeting higher criteria and are bound by greater commitments [10].

A. Cyber Rapid Response Teams and Mutual Assistance in Cyber Security

Declaration of Intent in the Field of Cyber Rapid Response Teams and Mutual Assistance in Cyber Security. The Ministers of Defence of the Member States and the Minister of National Defence of the Republic of Lithuania are welcoming the Joint Communication on cyber resilience, deterrence and defence adopted on September 13th, 2017.

Emphasizing opportunities in developing cyber projects through PESCO, they have expressed the intention:

- to develop and deepen voluntary cooperation in the cyber field through mutual assistance in response to major cyber incidents, including information sharing, joint training, mutual operational support, research and development and creation of joint capabilities;
- to create Cyber Rapid Response Teams, hereinafter referred to as the CRRTs, as a priority to provide mutual assistance between participating Member States (MS), and as appropriate to help other EU MS, EU institutions, including CSDP missions and operations, and eventually Partners to ensure higher level of cyber resilience and to respond to cyber incidents;
- CRRTs will complement national, EU, regional and multinational efforts in the cyber field, without duplicating existing efforts, structures and formats;

- to survey existing national and EU legal frameworks in order to investigate possibilities for an effective deployment of CRRTs and, if necessary, explore the need of their adaptation.

The Signatories intend to conclude detailed multilateral arrangements regarding the establishment and operation of the CRRTs project, including their mandate, tasks and responsibilities. The main elements of the CRRTs project should be the following:

Formation and cooperation: CRRTs should be formed by pooling participating MS experts on a rotational basis (including training and 6 months stand-by periods). CRRTs will be mobilised for a planned or urgent tasks agreed by all participating MS. CRRTs will only act upon the invitation from a MS, EU institution or Partner country;

Each CRRT should have a team leader and be composed of the participating MS cyber security experts (from Computer Security Incident Response Teams (CSIRTs)).

Designated experts will combine work in their original CSIRT and CRRT. CRRTs should closely cooperate with EU institutions, including CSIRT Network, European Union Agency for Network and Information Security (ENISA) and CERT-EU in order to ensure complementarity with existing cyber security initiatives.

The work of the CRRTs will be only within the scope, agreed by the MS.

Civil-Military nature: CRRTs would be a civil-military capability that should help foster civil-military culture in cyber domain and broaden cyber defence concept in the EU.

The civ-mil nature of CRRTs could also facilitate further cooperation between military and civilian CSIRTs. It is up to each MS to decide, which national CERT (civil or military) will participate in the project.

Equipment: In order to reach better operational capabilities of CRRTs, the Participants could explore and set the baseline of common Cyber Toolkits designed to detect, recognise and mitigate cyber threats.

To start operational activities CRRTs could use available on the market or nationally developed tools. However, to expand cyber security activities there is a need to develop a second generation unified deployable toolkit.

European Defence Fund co-funding and funding from other EU sources could be considered in this regard. It would facilitate industrial cooperation between participating MS and foster European cybersecurity industry.

Next steps: The Signatories intend to further shape and define the initiative, seeking to reach initial operational capabilities of CRRTs in 2019. The signatories intend to sign the Memorandum of Understanding by the end of 2019.

The Signatories participate on an equal basis in the process of creation of CRRTs.

The Ministry of National Defence of the Republic of Lithuania intends to be a lead nation of a project [11].

B. Progress of the EU CRRT Development

"We have achieved significant progress on the path of developing the EU cyber response force along with other participating countries and strategic partners of the project over the past year. Actions taken prove that we can develop and combine forces in practice to fight threats in cyberspace and to strengthen collective defence in this realm," says Lithuanian Deputy Defence Minister Edvinas Kerza.

In 2018 seven EU Member States have already signed Declaration of Intent to cooperate in creation of the EU CRRTs, and several more countries have voiced a wish to join in the future.

Moreover, Lithuania accomplished primary assessment of existing national legal frameworks among the participating Member States in order to investigate possibilities for an effective employment of EU CRRTs.

First steps to test the procedures were taken during cyber security exercise "Cyber Shield / Amber Mist 2018" in Lithuania, which involved national experts from participating EU Member States' cyber security units. The first EU CRRT for a 12-month rotation until 2020 was established and assigned by the Netherlands.

The EU CRRTs and Mutual Assistance in Cyber Security project initiative proposed back in 2017 is aimed at developing and deepening EU cooperation in the cyber field through mutual assistance in the response to major cyber incidents, incl. information sharing, joint training, mutual operational support, research and development (R&D) and creation technical toolkit.

Creation of EU CRRTs would strengthen EU Member States' capabilities to help each other, and as appropriate EU institutions, including CSDP operations, and would ensure higher level of cyber resilience and response to cyber incidents [10].

C. Current Development

Lithuania has reason to focus on cyber-threats: Though Lithuania saw a decrease in the number of total cyber incidents recognized last year, the country is reporting that the number of sophisticated cyber-attacks increased by 41 percent. Those malicious incidents include malware, system intrusion and compromised systems.

On average, the country sees 55,000 incidents each year — roughly 150 incidents a day — which is a lot when compared to population numbers in Lithuania, a government official told Defense News on an Atlantic Council-organized fact-finding mission in May. Presentations and meetings were conducted under the Chatham House Rule, so the individual cannot be identified.

Lithuania officially adopted a national cybersecurity strategy in August 2018, which is aimed to keep cyberspace for both the public and private sectors resilient against attacks.

Through the strategy, Lithuania has taken several unique routes in its approach to cybersecurity, including charging the Ministry of National Defence with the sole responsibility

of setting cyber policy. This has improved the organizational cybersecurity framework, the ministry reports.

The government is also building a secure state data-transfer network that is not linked to the public internet, which will be able to operate even if the public internet goes down, whether caused by a large-scale cyber-attack or a natural disaster, in order to secure data and maintain reliable communication.

The strategy also promotes the development of cyber defense capabilities, so in 2018, Lithuania, in cooperation with the United States, established a regional Cyber Security Center in Kaunas that will partially focus on research and development of next-generation capabilities, including the design of hardware to prevent cyber-attacks.

The Cyber Security Center is located in a nondescript office building adjacent to a busy skate park. It's still a work in progress, with much of the floor space empty and awaiting expansion, but a small room of staff were already busy at computers and 3D printers, building hardware and software tools aimed at preventing cyber-attacks.

One reason the center is focused on building its own hardware-based tools is because that approach is more secure, an official noted. The center has seven full-time staff members and two interns, and it is supported by the Pennsylvania National Guard, which will have personnel stationed at the center.

In addition to research and development, the center is also focused on training with partners and allies and conducting cyber-threat analysis.

The center is particularly interested in artificial intelligence and how that can enhance cybersecurity, real-time encryption and embedded-systems development, launching some of these R&D efforts last year and earlier this year.

By 2020, the center hopes to incorporate AI sensors into Lithuania's core governmental network.

Work from the center has already made the Lithuanian elections process more secure, an official reported.

Lithuania's focus on cybersecurity isn't just insular. The country participates at the international level by leading the EU's PESCO, project on cyber rapid response teams and mutual assistance in cybersecurity.

PESCO is part of the EU's security and defense policy that pursues structural integration among armed forces.

Common capability to respond to cyber-attacks by developing a cyber-toolkit that will support EU members in times of need are being created.

In addition to developing cybersecurity tools, conducting research and holding exercises, it is foreseen that the regional Cyber Security Center in Kaunas, Lithuania might also serve as a subdivision of the EU cyber rapid response teams.

But challenges remain, and they're not unique to Lithuania. The country acknowledges that technological solutions can't fully counter new vulnerabilities. Cyber-attacks against Lithuania are becoming more sophisticated, the Ministry of National Defence acknowledges, *and have been associated with hybrid threats*, where attacks were conducted in parallel to information warfare.

All of this is compounded by a growing number of devices in the country.

Social engineering methods are also being used actively and in different forms, the MoD reports, and the government networks and critical information infrastructure continue to be the most attacked sectors by malware [9], [17].

D. Directions of Future Research

Following the establishment of the Rapid Response Team as the first step, future research will focus on building opportunities and providing mutual assistance and cooperation in responding to major cyber incidents through information sharing, joint training, mutual operational assistance and creation of shared capabilities.

V. CONCLUSION

In respect of hybrid threats actions of strengthening cybersecurity in Europe, with wide-ranging concrete measures to provide a major boost to EU cybersecurity structures and response capabilities are being taken as well as tackling online disinformation to have a safer internet, to prevent interference in elections, and to provide better information about the EU and its policies.

In our paper we presented the novel constructed model of hybrid threats as well as the results of investigation of actions for cybersecurity and cyber defence in conditions of increasing challenge of cyber-attacks and the limited capabilities to respond to this threat. We described the process of creation and performance of EU Cyber Rapid Response Teams and Mutual Assistance in Cyber Security, introducing novel approach to cybersecurity and cyber defence at the EU level putting it into the context of the Hybrid Threats Model.

The project of developing the European Union Cyber Rapid Response Teams is well underway. Representatives of European Union member states participating in the projects met at the Ministry of National Defence on July 12th exchanged ideas and discussed the common cyber-toolkit to be developed.

The cyber-toolkit for the CRRTs will give the participating countries a capability for cyber-incident management. The meeting at the Ministry of National Defence addressed needs of every participant and the common vision. Representatives of the participating countries discussed possible funding mechanisms of the toolkit and its development plan.

The vision of the common cyber toolkit development is drafted and agreed by Estonian, Croatian, Polish, Dutch, Romanian and Finnish participants of the Lithuanian-initiated Cyber Rapid Response Teams and Mutual Assistance in Cyber Security project.

The toolkit will ensure the CRRTs have the basic technical equipment which is one of the factors for the lasting success of the project.

The Lithuanian-initiated Cyber Rapid Response Teams and Mutual Assistance in Cyber Security project is one of the most advanced projects developed in the framework of the PESCO.

Coordination of the project is led by experts of the Cyber Security and Information Technology Group of the Ministry

of National Defence [12]. In signing the Declaration the states announced their intent to:

- provide each other with assistance in response to major cyber incidents (including information-sharing, joint training, mutual operational support, R&D, and creation of joint capabilities)
- create Cyber Rapid Response Teams to provide this assistance to participating member states and, as appropriate, to other EU member states, EU institutions, and eventually Partners
- use the CRRTs to complement national, EU, regional and multinational efforts in the cyber field
- survey existing national and EU legal frameworks in order to ensure effective deployment of the CRRTs
- consider establishing Cyber Toolkits in order to improve CRRTs' operational capabilities, perhaps drawing on European Defence Fund co-funding and funding from other EU sources

The declaration also lays out some details about the CRRTs.

Each team is to have a designated leader, and be formed by pooling participating member state cybersecurity experts from Computer Security Incident Response Teams (CSIRTs) on a rotational basis, including training and six month standby periods. Designated experts will combine their work in their original CSIRT and their CRRT.

The teams will be mobilised for planned or urgent tasks agreed by all participating member states, and will only act at the request of a member state, EU institution, or Partner country. They are to cooperate closely with EU institutions, including the CSIRT Network, ENISA, and CERT-EU.

CRRTs are to be jointly civil and military initiatives, which "should help foster civil-military culture in cyber domain". The Declaration leaves it up to individual member states to decide which of its national CSIRTs (civil or military) will be involved in the project [3].

The first author of the paper participates in development of project on Cyber Rapid Response Teams and Mutual Assistance in Cyber Security as a team member.

REFERENCES

- [1] Chekinov, S. G. and Bogdanov, S. A.: "The Nature and Content of a New-Generation War." 2013
- [2] Council of the European Union: EU Cyber Defence Policy Framework, available at <https://www.consilium.europa.eu/media/37024/st14413-en18.pdf>, Accessed: October 20th, 2019.
- [3] Cyberviser: EU Cyber Rapid Response Teams project gets go-ahead, available at <https://www.cyberviser.com/2018/06/eu-cyber-rapid-response-teams-project-gets-the-go-ahead>, unpublished, Accessed: August 25th, 2019
- [4] European Union - EU: A Europe that protects: countering hybrid threats, available at https://eeas.europa.eu/sites/eeas/files/hybrid_threats_en_final.pdf, unpublished, Accessed: September 10th, 2019
- [5] Galinec, D.: Resilience Is Key: How to thwart known, and unknown, dangers. // per Concordiam: Journal of European Security and Defense Issues. Special Edition: Mission Critical; Protecting Essential Infrastructure from Cyber Attacks. Vol. 9, No. 1; pp. 14-21, 2018
- [6] Galinec, D., Možnik, D. and Guberina, B.: Cybersecurity and cyber defence: national level strategic approach, *Automatika*, Vol. 58, No. 3, pp. 266-272, 2017
- [7] Gerasimov, V. V.: Prevision is what Science is Valued For (Ценность науки в предвидении), *Voyenno-promyshlenniy kur'yer*, No. 8 (476), February 27th-March 5th, 2013 (in Russian)
- [8] Gorbachov, Y. Y.: A Cyber-War is Already on (Кибервойна уже идёт), *Nezavisimoye voyennoye obozreniye*, No. 13, April 12th-18th, 2013 (in Russian)
- [9] Judson, J.: A necessary rise: Lithuania bolsters its cybersecurity, catching the attention of other nations, *Fifth Domain*, available at https://www.eumonitor.eu/9353000/1/j4nvg5kjg27kof_j9vvik7m1c3gyxp/vkpldlchg8yqf/blg847046.pdf, Accessed: September 7th, 2019
- [10] Ltlife: Progress of the joint EU Cyber Rapid Response Teams development to be discussed in Amsterdam, available at <https://ltlife.lt/lt-life-english/progress-of-the-joint-eu-cyber-rapid-response-teams-development-to-be-discussed-in-amsterdam/>, Accessed: September 2nd, 2019
- [11] The Ministry of National Defence of the Republic of Lithuania: Declaration of Intent in the Field of Cyber Rapid Response Teams and Mutual Assistance in Cyber Security, available at https://www.eumonitor.eu/9353000/1/j4nvg5kjg27kof_j9vvik7m1c3gyxp/vkpldlchg8yqf/blg847046.pdf, Accessed: September 11th, 2019
- [12] The Ministry of National Defence of the Republic of Lithuania: Preparation of a memorandum on developing EU cyber response force begun in Vilnius, available at https://kam.lt/en/news_1098/current_issues/preparation_of_a_memorandum_on_developing_eu_cyber_response_force_begun_in_vilnius.html, Accessed: September 11th, 2019
- [13] NATO Special Operations Headquarters – NSHQ: Countering Hybrid Threats Seminar, Croatian Military Academy "Dr. Franjo Tuđman", Croatian Armed Forces, Ministry of Defence, Zagreb, September 3rd – 5th, 2019
- [14] Popov, I. M.: Network-Centric Warfare. Is Russia Ready for It? (Сетецентрическая война. Готова ли к ней Россия?), *Sovetskaya Rossiya*, January 19th, 2010 (in Russian)
- [15] Prokhvatilov V. V.: An Analysis of the 'Intelligent Web.' Metaprogramming, an organizational Weapon of the 21st Century. Indirect Action strategy (Атака мыслящей паутины. Метапрограммирование – организационные оружие XXI века), *Informatsionniye voyny*, Vol. 1, No. 13, pp. 73-83, 2010 (in Russian)
- [16] United States Department of Defense. Strategy for operating in cyberspace. Department of Defense, 2011
- [17] Szymoniak S., Siedlecka-Lamch O., Kurkowski M.: On Some Time Aspects in Security Protocols Analysis, in Proc.: International Conference on Computer Networks, Gliwice, Poland, Communications in Computer and Information Science book series, volume 860, pp. 344-356, Springer, 2018
- [18] Kollár, J., Florko, V.: Solution of selected problems using IBM QX, *Science & Military Journal*, No. 1, Vol. 14, 2019, pp. 5-10
- [19] Steingartner, W., Novitzká, V., Bačíková, M., Korečko, Š.: New Approach to Categorical Semantics for Procedural Languages, Computing and Informatics, Slovak Academy of Sciences, Bratislava, Slovakia, Vol. 36, No. 6, 2017, pp. 1385-1414

Modelling and computer simulations with the use of a specific organisation of calculations

Elzbieta Gawronska

Faculty of Mechanical Engineering and Computer Science

Czestochowa University of Technology

Czestochowa, Poland

gawronska@icis.pcz.pl

Abstract—Progress in computational methods has been stimulated by the widespread availability of cheap computational power leading to the improved efficiency of simulation software. Simulation tools become indispensable tools for engineers who are interested in attacking increasingly more significant problems or are interested in searching larger phase space of process and system variables to find the optimal design. In this paper, we show and introduce a new approach to modelling of a real physical process (binary alloy solidification) which involves time-stepping technique and allows to decrease computational cost. Implementation of our algorithm does not require a parallel computing environment but can use it after making some minor adjustments in source code. Our strategy divides domains of a dynamically changing physical phenomena. We are the first (to our best knowledge) to show that it is possible to use such a sequential organisation of calculations during the simulation of the solidification process. Our method is independent of domains considered, because of the natural separation of domain types (a cast, a core, a mould, etc.). Finally, we performed numerical experiments and demonstrate that our approach allows reducing computational time against traditional sequential computations and gives physically appropriate results.

Index Terms—modelling and computer simulations, applied informatics, sequential calculations, mixed time partitioning method

I. INTRODUCTION

Modelling and computer simulation become handy tools in studying difficult problems in foundry and metallurgical manufacturing. Computational investigations provide insight into the process that otherwise would be very difficult to obtain. For example, getting the complete temperature field evolution for given casting is an almost elementary step in computational studies, while experimentally it requires sophisticated tools to visualise the entire field. Moreover, computational studies are of high importance when there is a need to optimise casting production. Typical trial and error approach very often is insufficient, and more importantly, can be very laborious and expensive. Especially, when significant changes in the protocol need to be tested (e.g. several moulds, miscellaneous material properties or initial conditions). Usage of modelling tools, even a large series of studies can be performed without significant changes in the working space. Moreover, they allow

The project financed under the program of the Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2022 project number 020/RID/2018/19, the amount of financing 12,000,00 PLN

investigating the sensitivity of casts to changes in materials parameters or processing conditions.

To effectively perform a large series of computational studies using reasonable resources, the framework needs to be not only accurate but also computationally efficient. Combinatorial studies of phase space involve a large number of individual simulations. Moreover, increasing precision of simulation typically leads to more significant computational problems that need to be solved. In both cases, improving computational efficiency while providing high accuracy is of high importance, with several remedies already available. For instance, one option is to harness large computing clusters either based on CPUs [1], [2] or accelerated architectures such as GPUs [3], [4]. However, they require at least minimal expertise in high-performance computing, although possible may not be easily available. Another option is to explore more advanced computational techniques that minimise computational cost while providing an accurate solution.

In this work, we discuss one such technique, called mixed time partitioning method, where the considered structure is divided into sub-domains modelled by various procedures. This approach naturally fits the typical characteristics of solidification, where physical processes inside mould are of significantly different than those in a solidifying cast. Processes in casting are more dynamic than in mould, so they require a more accurate solution, i.e. small time step. On the other hand, a heat transfer within mould sub-domain is less intense, and thus coarse-grained step is sufficient to guarantee the desired precision of computations. As a result, different time steps and/or various time integration schemes can be chosen for individual sub-domains and considerable saving can be achieved. However, the adoption of suitable stability conditions is essential for accuracy. We discuss the principle of operation of our algorithm and the results of numerical simulations obtained by its use. Using of mixed time partitioning method increases the computational efficiency and gives promising results for future work.

II. METHODS

The casting is one of the typical techniques to fabricate metal machine elements and tools which properties depend on the interplay between many physical processes. Typically

there are complicated interaction phenomena, such as liquid-solid phase transformation, heat transfer including latent heat generation, solute diffusion, fluid dynamics and mould filling, just to name few. Understanding these interactions underpin the design of casts with desired properties. Computational techniques are a popular way to study these systems.

The problem of adequate modelling of foundry process, firstly, depends on the solution of heat transfer described by a partial differential equation. In a simple case, the equation can be analytically integrated with respect to space and time to yield accurate solution (see Fig. 1).

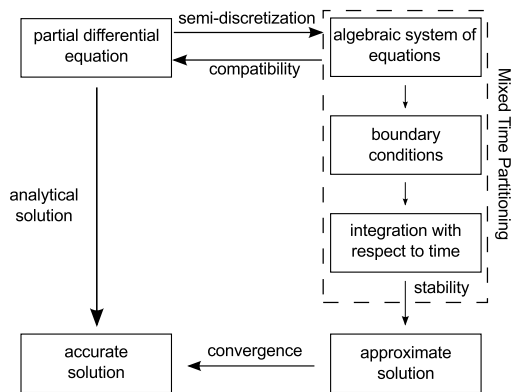


Fig. 1. Transition from partial differential equation to approximate solution by using mixed time partitioning methods.

However, owing to a complicated phenomenon, e.g. of solidification, it is impossible to carry out such a solution directly. In this case, it is necessary to perform numerical calculations which part can be mixed time partitioning method proposed.

We study time and space progress of temperature in the casting. The precision of computations is affected by two main factors: spatial discretisation (so-called semi-discretisation) and time discretisation. First of them is connected with size and type of finite elements, second is associated with the size of time step in integration schemes. In this work, we focus on the time discretisation. In particular, we check time steppers correctness by studying a combination of different time integration schemes used in various domains. We develop a new approach, that is a result of a merger two other techniques, namely sub-cycling and partitioning methods (see Figure 2). Such a combination allows better fitting the size of time step and time integration scheme for the modelled process.

Numerical model of solidification

Solidification is described by a quasi-linear heat conduction equation with extra source describing the rate of latent heat evolution along with boundary and initial conditions:

$$\nabla \cdot (\lambda \nabla T) + \rho_s L \frac{\partial f_s}{\partial t} = c \rho \frac{\partial T}{\partial t}, \quad (1)$$

$$\Gamma_1 : q = \alpha(T)(T_{\Gamma_1} - T_{env}(t)), \quad (2)$$

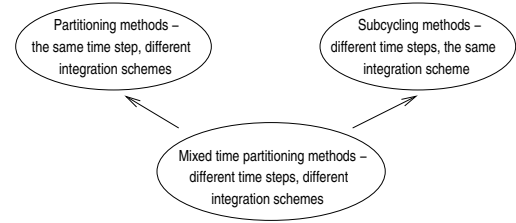


Fig. 2. Division scheme of mixed time integration methods.

$$\Gamma_2 : \begin{cases} -\lambda_1 \frac{\partial T^{(1)}}{\partial n} = -\lambda_2 \frac{\partial T^{(2)}}{\partial n} = \frac{\beta}{\delta} (T^{(1)} - T^{(2)}), \\ T^{(1)} \neq T^{(2)}, \end{cases} \quad (3)$$

where λ is the thermal conductivity coefficient, c is the specific heat, ρ is the density (subscript s refers to the solid phase), L is the latent heat of solidification, f_s is the solid phase fraction and T is the temperature. Boundary condition given by Eq. 2 is the Newton boundary condition, where the exchange of heat on Γ_1 with the environment takes place, with heat transfer coefficient α , where T_{Γ_1} is the temperature on the Γ_1 boundary of the domain and T_{env} is the environment temperature. Boundary condition given by Eq. 3 is the continuity boundary condition on the Γ_2 boundary separating Ω_1 and Ω_2 domains. The thermal conductivity coefficient of the material in the separating layer is given by β , and δ is the thickness of this layer and n is the normal vector to the boundary.

In this work, we use the apparent heat capacity formulation of solidification:

$$\nabla \cdot (\lambda \nabla T) = c^*(T) \frac{\partial T}{\partial t}, \quad (4)$$

$$c^* = c \rho_s, \quad \text{for } T < T_s,$$

$$c^* = c \rho_f - \rho_s L \frac{\partial f_s}{\partial T}, \quad \text{for } T_s \leq T \leq T_l, \quad (5)$$

$$c^* = c \rho_l, \quad \text{for } T > T_l.$$

where subscript l refers to the liquid phase, T_s is the solidus temperature (the beginning of solidification) and T_l is the liquids temperature (the end of solidification).

We used indirect model of solid phase growth for two-component alloy:

$$f_s(T) = \frac{1}{1 - nk\alpha} \left(1 - \left(\frac{T_M - T}{T_M - T_l} \right)^{\frac{1 - nk\alpha}{k - 1}} \right), \quad (6)$$

where n is the coefficient describing the grain shape, k is the solute partition coefficient, T_M is the solidification temperature of the pure basic component of alloy and α is the so-called Brody–Flemings coefficient. For more details we refer to [5], [6].

All the above equations form the basis of the thermal description of solidification. In general, to solve the partial differential equations, we have to pass from semi-discretisation through time discretisation to an approximate solution (see Fig. 1). First, we use the finite element method, successfully used

for numerical modelling of many phenomena, i.e. heat transfer, solidification, fluid flow, stress distribution, diffusion, and many more [7]–[12]. The semi-discretisation of the governing equation leads to the ordinary differential equation with time derivative, given as:

$$M(T)\dot{T} + K(T)T = b(T), \quad (7)$$

where M is the capacity matrix, K is the conductivity matrix, T is the temperature vector and b is the right-hand side vector, whose values are calculated on the boundary conditions basis.

Next, in respect of time-stepping schemes, we have studied a combination of two schemes from the family of one step Θ time integration schemes [13]:

- Euler Forward (conditionally stable)

$$M^n T^{n+1} = (M^n - \Delta t K^n) T^n + \Delta t b^n, \quad (8)$$

- modified Euler Backward (unconditionally stable)

$$(M^n + \Delta t K^n) T^{n+1} = M^n T^n + \Delta t b^{n+1}, \quad (9)$$

where superscript n refers following step of computations.

Mixed time partitioning methods

In this work, we propose to use a combination of two basic class of time-stepping schemes: explicit and implicit [13], [14]. Explicit methods usually are less computationally demanding, but small time step is required for numerical stability. Implicit methods, on the other hand, are more computationally expensive, but let much more significant time step, and in general, they stand out better stability. Enhanced stability of implicit methods is one of the reasons for their widespread use in many engineering problems. However, when the too large time step is used, despite numerical stability, implicit methods may give a solution with high, quickly propagate errors. The choice of the time-stepping scheme should be made based not only on the computational requirements but also the nature of modelled phenomena. For modelling of the intense changing process, like cast solidification, it is a good idea to use an explicit method with small time step allowing to precisely pick up for example moment of eutectic temperature achievement. Whereas, for not very dynamically process of heat transfer occurring in the mould, it seems reasoned using an implicit method with a larger time step. In problems like considered here solidification process, with two significantly different natures between casting and mould/cores, it may be beneficial to combine two time-stepping schemes. That is the primary motivation for this work and the basic idea of the mixed time partitioning method. To achieve this goal, we have implemented our method in C++ language supporting object-oriented techniques. Moreover, to additionally improve computational efficiency, we seek to decrease the size of the time step, only in sub-domains where it is justified. The choice of size of the time step is based on the numerical stability analysis, as we showed in [15], [16]. In Table I, we summarise the combinations of time integration schemes we examined.

One of the challenges in using mixed time partitioning method is related to the boundaries between domains with

different schemes. To avoid this obstacle, we introduce double numbering of nodes while keeping the same coordinates [6] (only on the boundary between domains). Thanks to this extra step, we get a natural separation of domains with different thermo-physical properties. Additionally, the double indexing helps to reorganise matrices and vectors in the corresponding system of equations. Finally, such operations allow to separate computations and to carry out simulations for sub-cycles (only for cast domain) and total-cycles (for all domains) independently.

Sub-cycles and total-cycles: The separation of domains into mould/cores and cast offers the possibility to use different time-stepping schemes and allows selecting the size of time steps in various sub-domains of the computational mesh. In particular, first, a small and fixed time step Δt in casting domain B is determined. Then, in mould sub-domain, A it is beneficial to choose a time step $m\Delta t$, where m is a positive integer value [16]. Calculations connected to cast domain are made in sub-cycles with time step Δt . Temperature vector is being updated only for elements belonging to casting domain. All elements in the mesh take part in calculations made in total-cycles with a time step $m\Delta t$. The local time step is always chosen according to stability requirements in given the sub-domain. It is important to note that the stability analysis is local and stability conditions of one sub-domain do not enforce the size of the time step for the entire mesh.

The proposed method requires to construct two systems of equations: (i) one for sub-cycles when the only cast is considered and (ii) one for total-cycle when mould/cores and cast are considered. In practice, global matrices and vectors are assembled according to the partitioning of the finite elements.

Solution algorithm: Calculations are made according to the following order (Algorithm 1). Computations connected with casting domain are always made as first at the beginning of simulations.

A counter c is responsible for the course of computations. It settles if total-cycle or sub-cycle is considered, and is defined as:

$$c = s \bmod m, \quad s = 1, 2, \dots, n_{step} \quad (10)$$

where n_{step} designates the number of computation steps during the simulation. Below, there is a fragment of source code illustrating a building of equations system in mixed time partitioning method.

```
CSysOfEqSolver* CSolidificationSolver::CreateSysOfEqSolver()
{
    CSysOfEqSolver* sys = 0;

    Index b = IsTotalCycle() ? GetNofNodes() : GetNofNodeSubcycle();

    if (CoefMatrixTraits::IsSparse())
        sys = new EqSolver(b, mIterations, mPrecision);
    else
        sys = new EqSolver(b, HalfBandWidth());
    return sys;
}
```

In turn, next two fragments of source code illustrate a modification of equations systems during inputting 3^h type

TABLE I
The combination of applied integration schemes

m	Θ_{cast}	Θ_{mould}	Δt_{cast}	Δt_{mould}	Symbol	Description
≥ 2	1	1	Δt	$m\Delta t$	$I - I^m$	Implicit sub-cycling — Implicit
≥ 2	0	0	Δt	$m\Delta t$	$E - E^m$	Explicit sub-cycling — Explicit
≥ 2	0	1	Δt	$m\Delta t$	$E - I^m$	Explicit sub-cycling — Implicit
≥ 2	1	0	Δt	$m\Delta t$	$I - E^m$	Implicit sub-cycling — Explicit

Algorithm 1 Organisation of computations.

```

set beginning conditions,
 $s \leftarrow 1$ ,  $c \leftarrow 1$ 
repeat
  if ( $c \neq 0$ ) then perform computations for sub-cycles:
    assemble the system of equations for elements in casting domain;
    input boundary conditions to the system of equations for casting domain;
    solve the system of equations for casting domain;
    compute temperature vector  $T^{n+1}$  for casting domain;
  end if
  if ( $c = 0$ ) then perform computations for full cycles:
    assembly system of equations for elements from casting and mould/cores domains;
    input boundary conditions for casting and mould/cores domains;
    compute temperature vector  $T^{n+1}$  for casting and mould/cores domains;
  end if
  increase computation step  $s \leftarrow s + 1$ ;
until ( $s < n_{step}$ );

```

of boundary condition:

```

void CThirdAHC_EF_BC::ModifySystemOfEquations(CSysOfEqSolver& pSys)
{
  const CSolidRegion& reg = static_cast<const CSolidRegion&>(Region());
  if (reg.Id() == regMould && !Solver().IsTotalCycle()) return;

  for (Index i = 1; i < GetNofNodes(); i++) {
    const CNode &n1 = Node(i-1), &n2 = Node(i);

    Real d = dist((CPoint&)n2, (CPoint&)n1);

    Real to = (temp(i-1)+temp(i))/2;
    Real a = (alpha(i-1)+alpha(i))/2 * d/6;

    Real ts = (reg.Id() == regMould) ?
      Solver().GetMultiple() * Solver().time_step() :
      Solver().time_step();

    Index i1 = n1.Id(), i2 = n2.Id();

    Solver().b(i1) += ts*a*(-2*Solver().u(i1) -
      Solver().u(i2) + 3*to);
    Solver().b(i2) += ts*a*(-Solver().u(i1) -
      2*Solver().u(i2) + 3*to);
  }
}

Real h = dist((CPoint&)n2, (CPoint&)n1);
Real ts = (reg.Id() == regMould) ?
  Solver().GetMultiple() * Solver().time_step() :
  Solver().time_step();
Real kapa = Kappa(i) * h/6 * ts;
Index i1 = n1.Id();
Index i2 = n2.Id();
Index i3 = Other()->Node(i-1).Id();
Index i4 = Other()->Node(i).Id();
Real u1 = Solver().u(i1);
Real u2 = Solver().u(i2);
Real u3, u4;
u3 = Solver().u(i3);
u4 = Solver().u(i4);
Solver().b(i1) -= kapa*(2*u1 + u2 - 2*u3 - u4);
Solver().b(i2) -= kapa*(u1 + 2*u2 - u3 - 2*u4);
}

```

RESULTS AND DISCUSSION

We have implemented the mixed time partitioning method in C++ programming language as an extension to the simulation framework called NuscaS [17]. We have used finite element method to solve enthalpy formulation of solidification, given by the Eq. 4, boundary conditions given by Eq. 2 and Eq. 3 and indirect model of solid-phase growth for two-component alloy given by Eq. 6. We have performed a numerical simulation of solidification for Al-2%Cu alloy casting solidifying in the metal form. The analysed casting together with mould and cores is shown in Figure 3.

and 4^h type of boundary condition:

```

void CFourthAHC_EF_BC::ModifySystemOfEquations(CSysOfEqSolver& pSys)
{
  const CSolidRegion& reg = static_cast<const CSolidRegion&>(Region());
  if (reg.Id() == regMould && !Solver().IsTotalCycle()) return;

  if (!OnSplitBound() || Node(0).Id() == Other()->Node(0).Id())
    return;

  for (Index i = 1; i < GetNofNodes(); i++) {
    const CNode &n1 = Node(i-1), &n2 = Node(i);

```

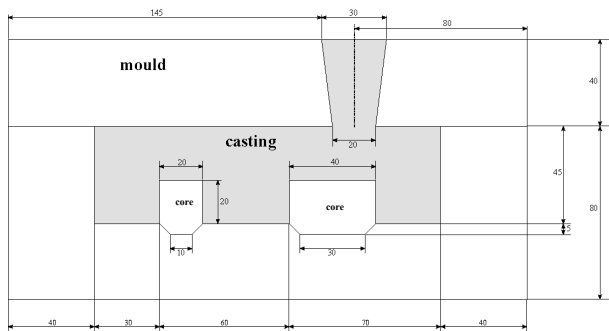


Fig. 3. The analysed casting in mould.

TABLE II
Physical properties of cast and mould material using during computer simulation of Al-2%Cu alloy solidification

Quantity	Unit symbol symbol	Liquid phase phase	Solid phase phase	Mould Mould
density	kg/m^3	2498	2824	7500
specific heat	J/kgK	1275	1077	620
thermal conductivity	W/mK	104	262	40

The values of material properties of a liquid phase, solid phase and mould/cores, used in the simulations, are shown in Table II.

Temperatures were taken from a phase diagram for the Al-Cu alloy system. The solidification heat is equal to $390 kJ/kg$ and the solute partition coefficient to be 0.125 (compare Eq. 6). The initial temperature of liquid alloy was $960 K$, while the temperature of mould was $590 K$ and the temperature of the cores were $540 K$. Non-ideal contact ($4^t h$ type boundary condition) was assumed between casting and mould (the conductivity of the separating layer was assumed to be equal to $1000 W/m^2K$) and between casting and the core (in this case the conductivity was assumed to be equal to $800 W/m^2K$). Neumann boundary conditions ($3^t h$ type) were assumed on external boundaries. It was assumed that environment temperature equals $300 K$ and heat exchange coefficient equals $100 W/m^2K$.

The analysed casting together with mould and cores is shown in Figure 3. The mesh was divided into 16261 triangular finite elements, obtaining 8640 nodes (with 2125 nodes in cast domain).

Calculations were carried out by using mixed time partitioning method, where $m = 15$ is the largest acceptable value and was taken from the numerical stability criterion. It was selected size of the time step $\Delta t = 0.003$ seconds to compare the results of the simulations for both explicit and implicit schemes. Calculations were carried out for 450 seconds duration of cast solidification and cooling process. After the time of 145 seconds, the cast was solidified in the whole area what means the solid phase fraction $f_s = 1$ in the entire area of casting.

To sum up, in Table I we have gathered various combinations of time integration schemes and sizes of time steps that

we have examined during numerical experiments of solidification. As can be seen the mixed time partitioning methods are similar to standard approach in only two particular cases ($E - E^m$ and $I - I^m$) when multiplication factor $m = 1$.

In this paper, the specific organisation and object-oriented implementation of calculations during solidification simulations are enunciated. Mixed time partitioning methods are proposed. These methods are a part of numerical computations of partial differential equations, and they combine different time integration techniques and size of time steps. To the best knowledge of the author, her work is the only study to numerical simulation of binary alloy solidification by using mixed time partitioning methods. What is more, the distribution of temperature and solid-phase fraction obtained for the multiplication factor $m = 15$ may confirm the correctness of used method and correspondence of computer simulations with physics of the solidification phenomenon.

In Figure 4, the temperature field in the casting is presented. This temperature field was obtained after 67.5 seconds of the duration of solidification processes.

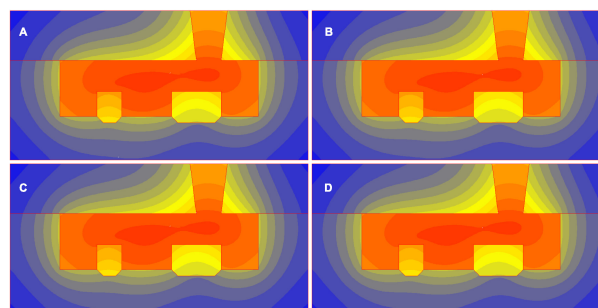


Fig. 4. Field of temperature in casting domain. Four combination of the time integration schemes a) $I - I^m$, b) $E - E^m$, c) $E - E^m$, d) $I - E^m$ for multiplication factor $m = 15$

In Fig. 5 and Fig. 6 the solid phase fraction distribution the map of solid phase fraction after 37.5 seconds is presented, respectively.

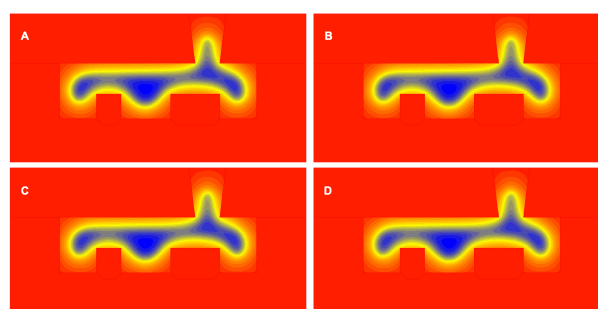


Fig. 5. Distribution of solid phase fraction in casting domain. Four combination of the time integration schemes a) $I - I^m$, b) $E - E^m$, c) $E - E^m$, d) $I - E^m$ for multiplication factor $m = 15$

In conclusion, it would seem that the computer simulations with the use of a specific organisation of calculations (mixed time partitioning method) gives possible to achieve accurate

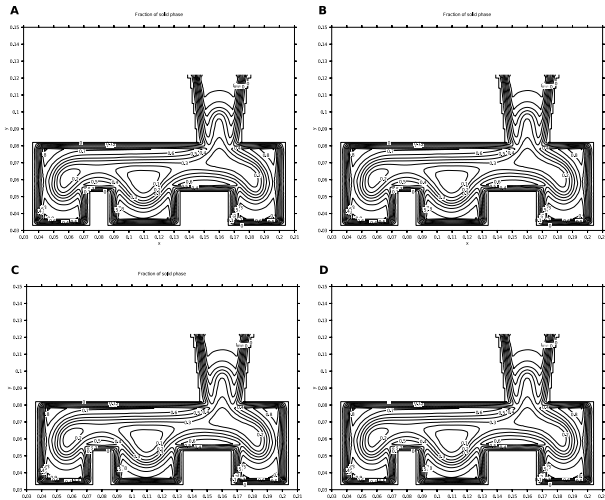


Fig. 6. Map of solid phase fraction in casting domain. Four combination of the time integration schemes a) $I-I^m$, b) $E-E^m$, c) $E-E^m$, d) $I-E^m$ for multiplication factor $m = 1$

results and can be applied on PC's available almost every foundry.

III. CONCLUSION

In this paper, we have presented a mixed time partitioning method with some exemplary of its implementation in C++ language with the use of object-oriented techniques. We have tested several variants with a different combination of time integration schemes, but also various choices of the size of the time step. Our results can give an improvement in the run-time of the proposed application compared with the standard form. In all cases, the precision of the computations was not affected.

We performed a series of tests and made several important observations:

- For the first time, we have shown that mixed time partitioning method a combination of different can be used. Moreover, we have shown that different time steps can be chased for different sub-domains (with conservation the stability condition).
- We can also conclude that the accuracy of the numerical solution is not affected (if stability conditions are satisfied).
- Finally, we note that the method presented here does not require specialised hardware or sophisticated algorithms. It can be relatively easy explored on desktop stations commonly available in foundry and metallurgical manufacture.

REFERENCES

- [1] J. Kim and R. Sandberg, "Efficient parallel computing with a compact finite difference scheme," *Computers & Fluids*, vol. 58, pp. 70–87, 2012.
- [2] R. Wyrzykowski, L. Szustak, and K. Rojek, "Parallelization of 2d mpdata eulag algorithm on hybrid architectures with gpu accelerators," *Parallel Computing*, 2014, in Press, Accepted Manuscript.
- [3] R. Strzodka, M. Dogger, and A. Kolb, "Scientific computation for simulations on programmable graphics hardware," *Simulation Modeling Practice and Theory*, vol. 13, pp. 667–680, 2005.
- [4] N. Yang, D. Li, J. Zhang, and Y. Xi, "Model predictive controller design and implementation on fpga with application to motor servo system," *Control Engineering Practice*, vol. 20, no. 11, pp. 1229–1235, 2012.
- [5] N. Sczygiol, "Approaches to enthalpy approximation in numerical simulation of two-component alloy solidification," *Computer Assisted Mechanics and Engineering Sciences*, vol. 7, pp. 717–734, 2000.
- [6] —, *Numerical modelling of thermo-mechanical phenomena in solidifying cast and casting mould*. Czestochowa: Czestochowa University of Technology, 2000, in polish.
- [7] M. Le Bars and M. G. Worster, "Solidification of a binary alloy: Finite-element, single-domain simulation and new benchmark solutions," *Journal of Computational Physics*, vol. 216, pp. 247–263, 2006.
- [8] C. Ye, J. Shi, and G. Cheng, "An extended finite element method (xfem) study on the effect of reinforcing particles on the crack propagation behavior in a metalmatrix composite," *International Journal of Fatigue*, vol. 44, pp. 151–156, 2012.
- [9] D. Wojtkowiak, K. Talaśka, I. Malujda, and G. Domek, "Estimation of the perforation force for polymer composite conveyor belts taking into consideration the shape of the piercing punch," *The International Journal of Advanced Manufacturing Technology*, vol. 98, no. 9, pp. 2539–2561, 2018.
- [10] E. Feulvarch, J. Roux, and J. Bergheau, "Finite element solution for diffusionconvection problems with isothermal phase changes," *Comptes Rendus Mecanique*, vol. 340, no. 7, pp. 512–517, 2012.
- [11] J. Olofsson and I. Svensson, "Incorporating predicted local mechanical behaviour of cast components into finite element simulations," *Materials & Design*, vol. 34, pp. 494–500, 2012.
- [12] D. Wilczyński, M. Berdychowski, D. Wojtkowiak, J. Grecki, and K. Wasa, "Experimental and numerical tests of the compaction process of loose material in the form of sawdust," *MATEC Web of Conferences*, vol. 254, p. 02042, 01 2019.
- [13] L. W. Wood, *Practical Time-stepping Schemes*. Oxford: Clarendon Press, 1990.
- [14] A. Gravouil and A. Combescure, "Multi-time-step explicit-implicit method for non-linear structural dynamics," *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 199–225, 2001.
- [15] T. Belytschko and Y. Y. Lu, "Convergence and stability analyses of multi-time step algorithm for parabolic systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 102, pp. 179–198, 1993.
- [16] E. Gawronska and N. Sczygiol, "Relationship between eugenvalues and size of time step in computer simulation of thermomechanics phenomena," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Hong Kong: Lecture Notes in Engineering and Computer Science, March 2014, pp. 881–885.
- [17] R. Dya and J. Mikoda, "3D simulation of alloy solidification in the NuscaS system," *Scientific Research of the Institute of Mathematics and Computer Scienc*, vol. 10, no. 1, pp. 33–40, 2011.

A Class of Multidimensional Oscillating Sequences and Related Young Functionals

Andrzej Z. Grzybowski

Institute of Mathematics

Czestochowa University of Technology

Czestochowa, Poland

andrzej.grzybowski@im.pcz.pl

Piotr Puchala

Institute of Mathematics

Czestochowa University of Technology

Czestochowa, Poland

piotr.puchala@im.pcz.pl

Abstract—In this paper, a class of sequences consisting of fast oscillating functions that are defined on multidimensional Euclidean space is defined. Next, some propositions providing probabilistic characterizations of Young measures connected with these sequences are stated. These propositions enable one to compute values of so-called Young functionals i.e. certain applications that assigns to any Carathodory function an integral with respect to related Young measure. These values are of primary interest in various contemporary non-convex optimization problems. Moreover, our results allow one to approximate the Young-functional's values by Monte Carlo simulations in all such cases where the integral cannot be computed explicitly. In this paper, a number of examples that illustrate possible applications of introduced formal results are provided as well.

Index Terms—Young measure, Young functional, fast oscillating functions, Monte Carlo simulations

I. INTRODUCTION

Non-convex optimization problems are at the core of various contemporary engineering applications. They arise in such areas as optimal control, microstructure theory or variational calculus.

One of the most important approaches to optimization problems is the so-called direct method where, roughly speaking, one constructs a sequence of functions $\{f_n\}$ that minimizes criterion-functional CF and then looks for the minimizer. Under some general assumptions about the criterion-functionals when the minimizing sequence is strongly convergent, i.e. it converges in the norm of the considered function-space, the minimizer is the function f_0 the sequence converges to (i.e. $\lim CF(f_n) = CF(f_0)$). It turns out however that if the sequence is not strongly convergent, then the optimization problems may not possess so understood classical minimizer. In particular, that situation concerns various oscillatory phenomena where the minimizing sequences $\{f_n\}$ are comprised of fast oscillating functions which converge to f_0 weakly but not strongly. Then in many interesting problems, $\lim CF(f_n) \neq CF(f_0)$, see e.g. [3]. An important question is: what is the nature of the "limiting" functional? The first answer was provided by L. C. Young. In his seminal work [7] he introduced a notion of "generalized curves" which are now called the "Young measures".

Nowadays we are presented with vast literature devoted to Young measures. There are various specific definitions of that

notion which depend on the considered function spaces as well as the assumed classes of functionals defined on them. This paper focuses on the Young measures related to some specific sequences of rapidly oscillating functions defined on a bounded subset of \mathbf{R}^d , while the considered functionals are defined via Carathéodory functions with domain in $\mathbf{R}^d \times \mathbf{R}^m$. The presented results are partly based on our paper [1] and they extend the results presented in [2].

In the next section, we introduce some preliminary definitions and results. In Section III we define certain classes of fast-oscillating sequences and we state a proposition that allows one to find explicitly the density functions of related Young measures in a number of interesting situations. Section IV illustrates possible applications of the main theoretical result stated in Section III along with discussion and comments on the presented examples.

II. PRELIMINARY DEFINITIONS AND RESULTS

Now we introduce basic notions of the Young functionals theory. The preliminaries presented here are based on books [3], [4], where the reader is referred to for more detailed introduction of the theory as well as necessary notions from functional analysis.

Let Ω be a nonempty, open and bounded subset of \mathbf{R}^d with smooth boundary. Denote by $L^\infty(\Omega)$ the Banach space of essentially bounded functions defined on Ω with values in a compact set $K \subset \mathbf{R}^m$. Let $\{f_n\}$ be a sequence of functions from $L^\infty(\Omega)$ and let $H(\cdot, \cdot): \Omega \times \mathbf{R}^m \rightarrow \mathbf{R} \cup \{\infty\}$ be given Carathéodory function, i.e. a function that is measurable with respect to the first and continuous with respect to the second variable. We consider the following integral functionals

$$CF(f_n) = \int_{\Omega} H(x, f_n(x)) dx \quad (1)$$

The problem we focus here on is the limit of the functionals' values when n tends to infinity. In the considered case, if the sequence $\{f_n\}$ is strongly convergent to certain function f_0 then $\lim CF(f_n) = CF(f_0)$. However if that sequence converges weakly but not strongly, then this nice property does not hold in general (unless the H is linear with respect to its second variable). A primary point of interest of the theory is the nature of that limit in such cases. To formulate

important general result concerning this issue we need to look at the integral (1) from yet another side. For any given Borel function f_n we can treat the integral (1) as a linear functional I_n defined on the space \mathcal{H} of the Carathéodory functions $H(\cdot, \cdot): \Omega \times \mathbf{R}^m \rightarrow \mathbf{R} \cup \{\infty\}$. Thanks to Banach-Alaouglu-Bourbaki theorem we know, that there is a subsequence of the functionals $\{I_n\}$ that is weakly* convergent to a certain functional - say YF - on the space \mathcal{H} . To simplify our considerations and notation, we assume from now on that simply the sequence $\{I_n\}$ is weakly* convergent on \mathcal{H} . The nature of the functional YF is really interesting, and in the considered situation it is revealed by the following theorem (compare Theorem 6.2 in [3])

Theorem 2.1: Let $\Omega \subset \mathbf{R}^d$ be a measurable set and let $f_n: \Omega \rightarrow K$ be measurable functions such that

$$\sup_n \int_{\Omega} h(|f_n|) dx < \infty,$$

where $h: [0, \infty) \rightarrow [0, \infty)$ is a continuous, nondecreasing function such that $\lim_{t \rightarrow \infty} h(t) = \infty$. There does exist a family of probability measures $\nu = \{\nu_x\}_{x \in \Omega}$ depending measurably on x with the property that whenever the sequence $\{H(\cdot, f_n(\cdot))\}$ is weakly convergent in $L^1(\Omega)$ for any Carathéodory function H , then the weak limit is the function \overline{H} given by

$$\overline{H}(x) = \int_K H(x, \lambda) d\nu_x(\lambda).$$

■ In the case that is considered in Theorem 2.1, the weak convergence of $\{H(\cdot, f_n(\cdot))\}$ simply means the convergence "in average" in Ω , i.e. for any Carathéodory function H

$$\int_{\Omega} H(x, f_n(x)) dx \xrightarrow{n \rightarrow \infty} \int_{\Omega} \overline{H}(x) dx \quad (2)$$

As a consequence we learn the nature of the weak* limit of the sequence $\{I_n\}$. The limiting functional YF is given by the right-hand side of (2). It is called the *Young functional*. The family of probability measures $\nu = \{\nu_x\}_{x \in \Omega}$, the existence of which is assured by Theorem 2.1, is called the *Young measure* associated with the sequence $\{f_n\}$.

Sometimes the Young measure $\nu = \{\nu_x\}_{x \in \Omega}$ does not depend on $x \in \Omega$. In this case we denote it merely by ν . Such Young measure is called the homogeneous one.

III. MULTIDIMENSIONAL RAPIDLY OSCILLATING SEQUENCES WITH UNIFORM REPRESENTATION

Let function $f: \Delta \rightarrow K \subset \mathbf{R}^m$ be a Borel function defined on the d -dimensional box (hyperrectangle) $\Delta = [a_1, b_1] \times \dots \times [a_d, b_d]$ $b_i > a_i$, $i = 1, \dots, d$, and let $f^e: \mathbf{R} \rightarrow K$ be the periodic extension of f i.e. the following condition holds:

$$f(x_1, \dots, x_d) = f^e(x_1 + k_1 T_1, \dots, x_d + k_d T_d)$$

for arbitrary integers k_i and $T_i = b_i - a_i$.

Let Ω be a given d -dimensional box. A sequence $\{f_k\}$ of functions $f_k: \Omega \rightarrow K$, $k = 1, 2, \dots$ defined by the formula

$$f_k(x) = f^e(kx), \quad x \in \Omega \quad (3)$$

will be called a Multidimensional Rapidly Oscillating Sequence with Uniform representation f , and denoted as $\text{MROSU}(f)$. We will also say that f generates rapidly oscillating sequence $\{f_k\}$. Note that $x = (x_1, \dots, x_d)$ and $f^e(kx) = f^e(kx_1, \dots, kx_d)$. It should be also emphasized that the Ω - the domain of elements of $\text{MROSU}(f)$ - does not have to be the same as Δ i.e. the domain of f .

Example 1 In this example we present illustrative plots of some elements of $\text{ROSU}(f)$, where $f: \Delta \rightarrow K \subset \mathbf{R}$

$$f(x_1, x_2) = \frac{5}{4} - (x_1 - 1)^2 - (x_2 - 1)^2, \quad (4)$$

and $\Delta = [0, 2] \times [0, 2]$

The behavior of MROSU in this case is illustrated by the plots of the function f given by (4) as well as of exemplary elements of $\text{MROSU}(f)$ with the domain $\Omega = \Delta$. They are presented in Fig.1. Namely it shows the plots of $f = f_1, f_2, f_4$ and f_{10} .

It is seen that the graphs of f_k are getting denser when k tends to infinity. Unfortunately, a conventional weak limit of $\{f_k\}$ loses most of the information about the fast oscillations in $\{f_k\}$ because it takes into account only the mean values of $\{f_k\}$. That is why we need a new concept of the limit and here the theory of Young measures helps us. If ν_x is the Young measure associated with $\{f_k\}$ then, roughly speaking, for any measurable set $A \subset K$ the intuitive meaning of $\nu_x(A)$ is the probability that for an *infinitesimally* small neighborhood S of $x \in \Omega$ and sufficiently large k 's we can "find" $f_k(s)$ in A , when s changes within S . Obviously it depends on the character of the oscillations in the sequence.

A. Young measures associated with the $\text{MROSU}(f)$

Directly from the definition of $\text{MROSU}(f)$ we see that when k tends to infinity its behavior is exactly the same in every neighborhood of any $x \in \Omega$. More precisely, its asymptotic behavior in an arbitrarily small box $B \subset \Omega$ does not depend on the box position within the domain. Consequently, it is obvious that the Young measure associated with the $\text{MROSU}(f)$ is the homogeneous one.

Now we formulate our main result concerning considered class of oscillatory sequences.

Proposition 3.1: Young measure generated by $\text{MROSU}(f)$ is the homogeneous one. This measure is identical with the probability distribution of the random vector $Y = f(U_{\Delta})$, where U_{Δ} is a random vector uniformly distributed on Δ .

The proof of the above fact is similar to the proof of analogous result concerning the 1-dimensional case and which is presented in [1]. Now we show some possible applications of the Proposition 3.1.

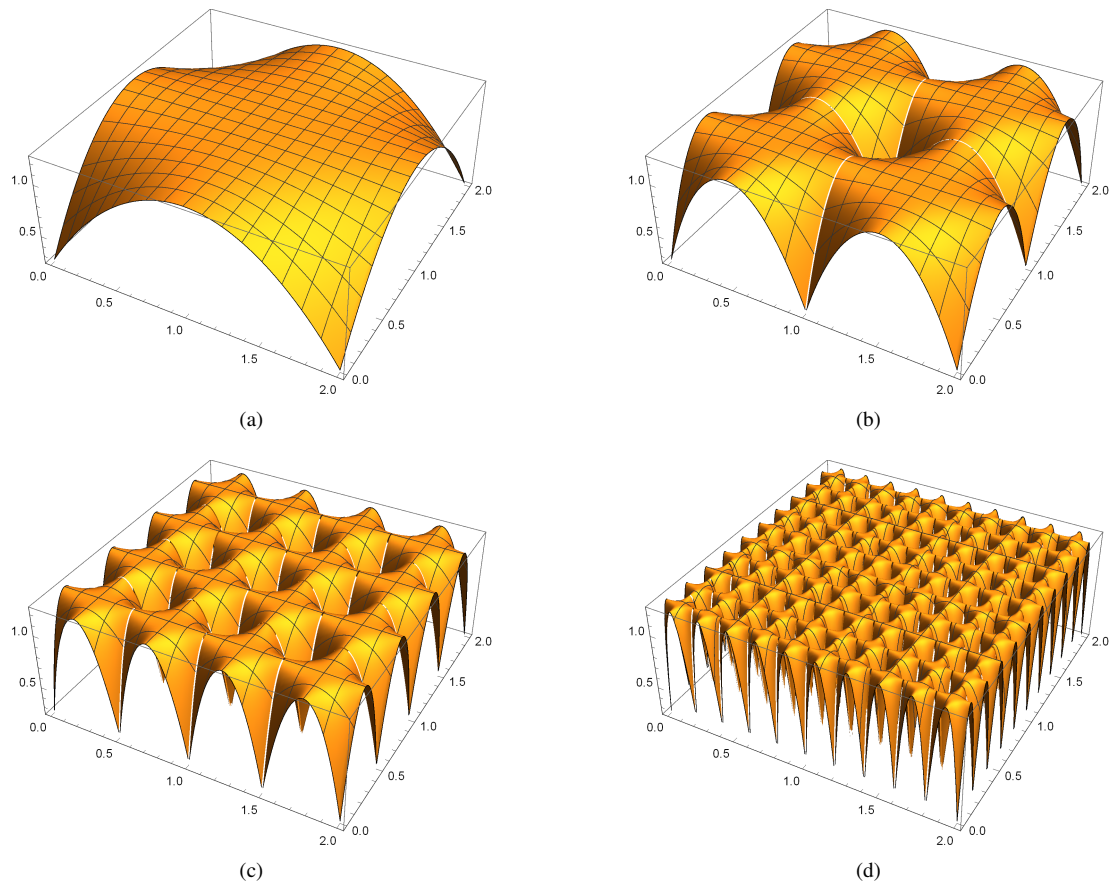


Fig. 1: A function f given by Eq. (4) and functions f_2, f_4 and f_{10} belonging to $\text{MROSU}(f)$ with the domain $\Omega = [0, 2) \times [0, 2)$. Because in this case $\Omega = \Delta$ the functions f and f_1 are the same. Plots of the functions $f = f_1, f_2, f_4$ and f_{10} are labeled as a, b, c, and d, respectively

IV. APPLICATIONS AND DISCUSSION

Due to the above general characterization of the Young measures generated by $\text{MROSU}(f)$ one can find explicit forms of their probability density functions in various specific situations. It is because the probability theory provides us with a number of results concerning the distributions of random vectors $Y = f(U)$ in relation to the distribution of the vector U and the conditions imposed on the multidimensional vector function f .

Let us for example consider the case where $d = m$ and the Borel function f satisfies the following conditions:

- f is the one-to-one mapping from Δ to K
- all components of f have continuous partial derivatives with respect to all their arguments
- the Jacobian determinant $J_f(x)$ of f does not vanish at any point $x \in \Delta$

Let $h = (h_1, h_2, \dots, h_m)$ and $J_h(\cdot)$ denote the inverse of f and its Jacobian determinant.

In such a case well-known probabilistic result concerning the probability distributions of such vector mappings allow us to state the following corollary to the Proposition 3.1 (compare Theorem 5.4 in [5]).

Corollary 4.1: Under the above stated assumptions, the Young measure generated by the $\text{MROSU}(f)$ is a homogeneous one and its probability density function g with respect to the Lebesgue measure on K has the following form

$$g(y) = \frac{1}{\mu(\Delta)} \mathbf{1}_\Delta(h_1(y), \dots, h_m(y)) |J_h(y)| \quad (5)$$

where $\mu(\Delta)$ and $\mathbf{1}_\Delta$ are, respectively, the Lebesgue measure and the characteristic function of Δ .

The above corollary can be also used for the cases where $d > m$, because then we can often enlarge the dimension of the Y in such a way that the new "artificial" vector function satisfies the necessary assumptions. Then the joint probability distribution of the Y alone can be obtained as the marginal distribution of the artificially extended vector.

Now we apply the corollary in order to compute values of Young functionals related to some exemplary $\text{MROSU}(f)$.

Example 2 Let us consider the $\text{MROSU}(f)$, defined on $\Omega = (0, 2) \times (0, 3)$ by a function $f: \Delta \rightarrow K \subset \mathbf{R}$

$$f(x_1, x_2) = x_1 x_2^2 \quad (6)$$

with $\Delta = [0, 1) \times [0, 1)$ The plots of the function f and the first element of $\text{MROSU}(f)$ are shown in Fig. 2. We are interested

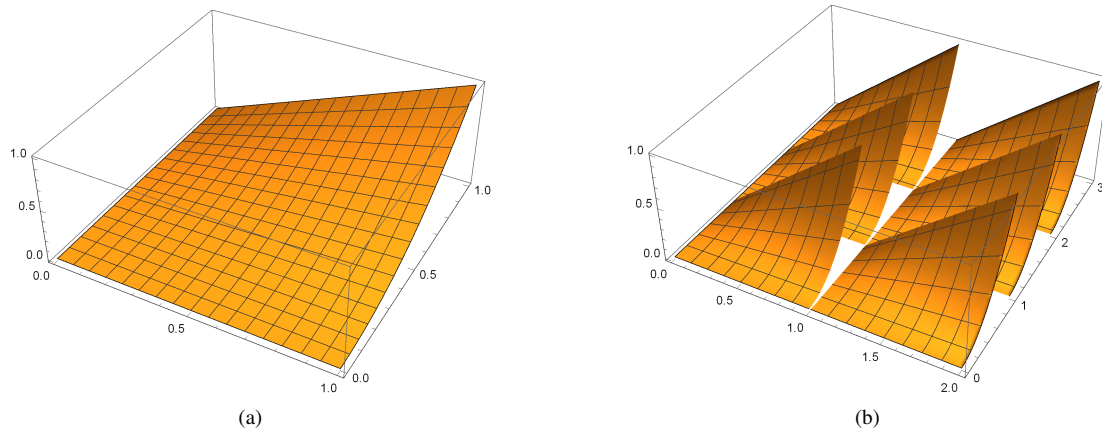


Fig. 2: A function f given by Eq. (6) and the element f_1 , belonging to $\text{MROSU}(f)$. The domain of f is $\Delta = [0, 1] \times [0, 1]$, whilst domain of the MROSU is $\Omega = [0, 2] \times [0, 3]$. Plots of the functions f , f_1 are labeled as a, b, respectively

in the limits :

$$\lim_{k \rightarrow \infty} \int_{\Omega} H(x_1, x_2, f_k(x_1, x_2)) dx_1 dx_2 \quad (7)$$

for any Carathéodory function $H : \Omega \times K \rightarrow \mathbf{R}$, and f_k given by Eq. 3

For this purpose we can compute the density function of the Young measure associated with the MROSU and make use of the Eq. (2). It can be done with the help of Corollary 5. A little calculus allows us to find the following probability distribution function of the related Young measure:

$$g(y) = \frac{1 - \sqrt{y}}{\sqrt{y}} \mathbf{1}_{(0,1)}(y) \quad (8)$$

Now that we have the density function g , it is easy to compute the function \bar{H} (see Theorem 2.1) as:

$$\bar{H}(x_1, x_2) = \int_{(0,1)} H(x_1, x_2, y) g(y) dy.$$

and then the limits (7) as the integrals (2), i.e.

$$\int_{\Omega} \bar{H}(x_1, x_2) dx_1 dx_2$$

Below we present some exemplary results:

- 1) for $H(x_1, x_2, y) = x_1 + x_2 y^2$:
 - $\bar{H}(x_1, x_2) = x_1 + x_2/15$
 - the limit (7) equals $33/5$
- 2) for $H(x_1, x_2, y) = \log(x_1 + x_2 + y)$:
 - $\bar{H}(x_1, x_2) = -3 + 4\sqrt{x_1 + x_2} \operatorname{arccot}(\sqrt{x_1 + x_2}) + \log(1 + x_1 + x_2) - (x_1 + x_2) \log(1 + \frac{1}{x_1 + x_2})$
 - the limit (7) equals $\frac{1}{30}(-594 - 48\sqrt{3}\pi - 128\sqrt{2} \operatorname{arccot}(\sqrt{2}) + 800\sqrt{5} \operatorname{arccot}(\sqrt{5}) - 296 \log 2 - 390 \log(3) + 625 \log(5))$
- 3) for $H(x_1, x_2, y) = x_1 \cos(x_2 y)$:
 - $\bar{H}(x_1, x_2) = \frac{x_1}{x_2} (\sqrt{2\pi y} FC(\sqrt{2x_2/\pi}) - \sin(x_2))$, with $FC(\cdot)$ being the Fresnel C integral

- the limit (7) equals $4\sqrt{6\pi} C(\sqrt{6/\pi}) 2 \sin(3) + \operatorname{Si}(3)$ with $\operatorname{Si}(\cdot)$ being the sine integral.

□

Our Proposition 3.1 is useful also in those cases where the corollary cannot be adopted to obtain the density function of the associated Young measure, e.g. when the f is not the one-to-one mapping as it is in Example 1. However, in such cases the Monte Carlo simulations can be used to obtain the Young functional values.

Monte Carlo methods has long history that starts during Second World War - they were developed by Stanislaw Ulam during the famous Manhattan Project. Nowadays Monte Carlo simulation is both a powerful engineering tool and very important source of scientific knowledge, [6]. In particular, Monte Carlo simulations can be used to evaluate values of integrals in various situations. Thus here, we adopt this approach to evaluate Young functional $\text{YF}(H) = \int_{\Omega} (\int_K H(x, y) d\nu(y)) dx$.

By Proposition 1, the Young measure ν is the probability distribution of the random variable $Y = f(U_{\Delta})$. So, we may look at the Young-functional-value as the expected value of the random variable $\int H(x, Y) dx$ and it can be estimated by its empirical mean (i.e. the arithmetic mean of its randomly generated sample). Thus, in case where the Δ is a Cartesian product of intervals I_k , $k = 1, \dots, d$ one can use the following procedure YFE to evaluate values of such Young functionals:

- Step 0 Set $k = 1$
- Step 1 Set $t = (\text{Random}(I_1), \dots, \text{Random}(I_d))$
- Step 2 Set $y = f(t)$
- Step 3 Set $z[k] = \text{NINT}(H(x, y), x, \Omega)$
- Step 4 N times repeat Steps 1 to 3
- Step 5 Set $\text{sample} = (z[1], \dots, z[N])$
- Step 6 Set $\text{YF} = \text{Mean}(\text{sample})$
- Step 7 Return YF

The above procedure is called with the following arguments: the formula that defines the Caratheodory function H, the formula for the function f that defines the MROSU and its

domain $\Delta = I_1 \times \dots \times I_d$, the domain Ω of the ROSU itself, and the size N of the sample that will be used to evaluate the Young functional's value. The subroutine $Random(I)$ returns a pseudorandom number generated according to the uniform probability distribution defined on an interval I . The subroutine $NINT(H(x,y), x, \Omega)$ returns the result of the numerical integration $\int_{\Omega} H(x,y) dx$. Obviously the numerical integration is possible, because when the subroutine is called, the value of y is already fixed, as it is set up at Step 2.

Now we are able to consider the MROSU from Example 1. For instance let us evaluate the value of the related Young functional for the Caratheodory function $H(x_1, x_2, y) = x_1 + x_2 \log(y)$. For that purpose we have implemented the above procedure in Wolfram Mathematica 10.5.

The result obtained for the sample size 5000 is 101.488. However, in this case we do not know the magnitude of the estimation error. The usefulness of the Monte Carlo simulations in computation of the values of the Young functionals can be verified by their comparison with the correct results found with help of Young measure. So, to check the correctness of the YFE procedure results, let us go back to Example 2. These comparison results are presented in Table I. That table compares the values of Monte Carlo approximations with the exact values presented in Example 2. We see that the approximations are really precise.

To approximate the Young functional's values one may also directly integrate the sufficiently distant elements of the converting sequence, i.e. one may compute $\int_{\Omega} H(x_1, x_2, f_k(x_1, x_2)) dx_1 dx_2$ for sufficiently large k . The exemplary results of such approach are presented in Table II. As we see the evaluation errors are much bigger when we try to approximate limit (7) with the help of the distant elements of the sequence than those related to Monte Carlo evaluations. What is more, the absolute relative error does not decrease monotonically as we increase the sequence-element's number. On the contrary - in our example, the errors are the greatest for the largest considered k . It is because the integration of fast oscillating functions is a highly difficult numerical task. What is also worth emphasizing, the quality of the numerical integration would be even worse in the higher dimensions, whilst the results obtained via the Monte Carlo approximation are much more precise. It is because the precision of Monte Carlo approximation is very robust against the changes in dimensions of the domains.

REFERENCES

- [1] A.Z. Grzybowski, and P. Puchala, "On Young Functionals Related to Certain Class of Rapidly Oscillating Sequences", *IAENG International Journal of Applied Mathematics*, vol. 48, no. 4, pp 381-386, 2018
- [2] A.Z. Grzybowski, and P. Puchala, " Monte Carlo Simulation in the Evaluation of the Young Functional Values", *Proceedings of IEEE 14th International Scientific Conference on Informatics, Poprad*, (eds. Novitzka, S. Korecko, A. Szakal), New York 2017, pp221-226
- [3] P. Pedregal, *Variational Methods in Nonlinear Elasticity*, SIAM, Philadelphia 2000.
- [4] T. Roubíček, *Relaxation in Optimization Theory and Variational Calculus*, Walter de Gruyter, Berlin, New York, 1997.

TABLE I: Results Of Monte Carlo evaluations of the Young functional's values - Example 2

Caratheodory integrand	Sample size	Estimated value	Relative error
$H(x_1, x_2, y) = x_1 + x_2 y^2$	5000	6.5967	0.0005
	10000	6.6042	0.0006
	20000	6.6099	0.0015
$H(x_1, x_2, y) = \log(x_1 + x_2 + y)$	5000	5.2597	0.0010
	10000	5.2657	0.0002
	20000	5.2632	0.0003
$H(x_1, x_2, y) = x_1 \cos(x_2 y)$	5000	5.4713	0.0018
	10000	5.4798	0.0002
	20000	5.4856	0.0008

TABLE II: Results of the numerical integration of exemplary elements of the sequence (7) - Example 2

Caratheodory integrand	Element number	Estimated value	Relative error
$H(x_1, x_2, y) = x_1 + x_2 y^2$	$k=100$	6.5395	0.0092
	$k=500$	6.6187	0.0028
	$k=1000$	6.5782	0.0033
$H(x_1, x_2, y) = \log(x_1 + x_2 + y)$	$k=100$	5.2580	0.0013
	$k=500$	5.2500	0.0028
	$k=1000$	5.2393	0.0048
$H(x_1, x_2, y) = x_1 \cos(x_2 y)$	$k=100$	5.5473	0.0121
	$k=500$	5.5112	0.0054
	$k=1000$	5.5519	0.0129

- [5] T.T. Soong, *Fundamentals of Probability and Statistics for Engineers*, John Wiley and Sons, Chichester, England, 2004.
- [6] [168] Winsberg E.B., *Science in the Age of Computer Simulation*, The University of Chicago Press, Chicago, 2010. ISBN-13: 978-0-226-90204-3
- [7] L.C. Young, "Generalized curves and the existence of an attained absolute minimum in the calculus of variations," *Comptes Rendus de la Société des Sciences et des Lettres de Varsovie, classe III* vol. 30, pp. 212-234, 1937.

Comparative study of inconsistency indices based on a notion of a triad inconsistency

Andrzej Z. Grzybowski

Institute of Mathematics

Czestochowa University of Technology

Czestochowa, Poland

andrzej.grzybowski@im.pcz.pl

Tomasz Starczewski

Institute of Mathematics

Czestochowa University of Technology

Czestochowa, Poland

tomasz.starczewski@im.pcz.pl

Abstract—Many prioritization methods are based on a pairwise comparison matrix (PCM). In the applications of the PCM, a special attention is paid to the PCM inconsistency analysis. To characterize this inconsistency some indices are proposed. All inconsistency indices proposed in literature share two features: they are nonnegative and equal 0 if and only if a given PCM is consistent. From purely mathematical point of view we have infinitely many nonnegative matrix functions that satisfy such conditions. In this paper we propose a new class of inconsistency indices that is based on a notion of a triad inconsistency. Their performance is compared with the help of Monte Carlo experiments. In these experiments the criterion for the indices performance is their relationship with the final-priority-estimates-errors.

Index Terms—prioritization, pairwise comparisons, inconsistency indices, Monte Carlo simulations

I. INTRODUCTION

Multiple-criteria decision analysis (MCDA) deals with decision-making problems that have only a small number of alternatives that need to be ranked. In all these problems the decision-maker creates a ranking of the available decision alternatives. In the MCDA it is usually achieved by estimation of the priority weights, i.e. numbers that tell to what degree a given alternative satisfies the given criterion. Apart from the alternatives and a number of criteria, the more complex MCDA problems may also take into account other factors such as experts and/or decision-makers (DM). In order to obtain the final ranking of the available alternatives all factors have to be ranked - so the ranking creation, i.e. the prioritization, is the essence of the MCDA. Recent literature, e.g. [17], claim that the most frequently used prioritization methodology is the Analytic Hierarchy Process (AHP), [14]. Under the AHP framework, the final alternatives' ranking is created after all sub-problems of prioritization are solved (i.e. the rankings of DMs, experts, criteria, and alternatives with respect to every single criterion are obtained). In the AHP, the basis for any of these rankings creation is formed by the so-called pairwise comparison matrix (PCM). This matrix contains the decision maker's judgments about the priority-weights-ratios. Because of human brain limitations, it is natural that the DM's judgments are usually erroneous. Thus some level of judgments' incorrectness has to be accepted. On the other hand, it is obvious that serious errors in the DM's judgments make the

information contained in the PCM useless. Consequently, an important problem within the AHP methodology is the ability to distinguish between useful PCMs and useless ones. So one of the fundamental questions connected with this methodology is how to measure the degree of inconsistency of the PCM. To characterize the degree of such inconsistency some inconsistency indices are introduced in literature. However the usage of these indices is justified only by some heuristic arguments - it is not known what the indices really tell us about the usefulness of the PCM at hand. All these inconsistency indices share only two features: they are nonnegative and equal to 0 if and only if a given PCM is consistent. However, looking at this problem from a mathematical perspective, we have infinitely many nonnegative matrix functions that satisfy such conditions, and all of them are - a priori - possible candidates for inconsistency index. In this paper, we propose a new class of inconsistency indices that are based on a triad inconsistency, a notion introduced by Koczkodaj in [12]. Then in order to compare the performance of the inconsistency indices we adopt an approach introduced in [6], where it is proposed to examine the relationship between their values and the magnitude of the prioritization errors. In our experiments, we simulate the typical AHP setup consisting of decision alternatives and a number of different criteria. Such an approach for the first time in literature was suggested in [11].

This paper is organized as follows. Section 2 introduces all necessary notions and definitions. In Section 3 the considered problem is formally stated. Section 4 describes the adopted simulation framework and presents the results of the simulation experiments along with discussion and conclusions.

II. NOTATION, NECESSARY DEFINITIONS AND BASIC FACTS

A priority vector (PV) is a vector $\mathbf{v} = (v_1, \dots, v_n)'$, consisting of priority weights i.e. numbers that reflect the intensity of importance of each alternative with respect to a given criterion. Typically, the priority weights v_i , $i = 1, \dots, n$, are positive and the PV is normalized to unity, [16].

As a result of comparisons of each pair of alternatives, the pairwise comparison matrix $\mathbf{A} = [a_{ij}]_{n \times n}$, with $a_{ij} = v_i/v_j$, is obtained. In the conventional AHP, the input data of the PCM is collected only for the upper triangle of the matrix

\mathbf{A} , while the remaining elements are computed as the inverses of the corresponding symmetric elements in the upper triangle i.e. $a_{ij} = 1/a_{ji}$ for $i > j$. A PCM that satisfies this condition is said to be *reciprocal*. A PCM is called a *consistent* one, if it is reciprocal and its elements satisfy the condition:

$$a_{ij}a_{jk} = a_{ik} \quad \forall i, j, k = 1, \dots, n$$

Such a PCM is a basis for the priority weights estimation. There are two most popular prioritization methods. One of them is the geometric mean method - GM, [4]. The estimated priority vector (EPV) in the GM can be obtained by the following formula:

$$w_i = \left(\prod_{j=1}^n a_{ij} \right)^{1/n} / \sum_{j=1}^n \left(\prod_{i=1}^n a_{ij} \right)^{1/n}$$

Even more commonly used prioritization method is based on specific results from the spectral theory. This one is called right eigenvector method (REV). Its description and formal backgrounds have a vast literature, see e.g. [14], [15] or many university textbooks devoted to the AHP. In short, the REV method suggests to take as the EPV, the normalized eigenvector associated with the principal eigenvalue of the PCM at hand. To achieve this, one has to solve the characteristic equation

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

and then find a normalized vector \mathbf{w} that satisfies the equation

$$\mathbf{A}\mathbf{w} = \lambda_{\max} \mathbf{w}$$

where λ_{\max} is the principal eigenvalue of \mathbf{A} . The Perron's theorem assures that such a definition of the EPV makes sense (e.g. the theorem assures the uniqueness of \mathbf{w} and positivity of all its coordinates).

A number of current papers examine the performance of these and other prioritization methods. In those studies it is found, e.g. [5], [3], [13], that both the GM and REV result in very similar EPVs - they differ very little and it is not known which one is better - both of these methods have their supporters. Nonetheless, the GM is much easier to use, thus this method will be adopted in presented here studies.

Now let us go back to the issue of DM's judgments expressed in the PCM. In praxis, one cannot expect that the elements of PCM give the priority ratios precisely. First of all, typically, the DM's answers are expressed in numbers belonging to a given, predefined scale. So, one cannot neglect rounding errors. For example, the most popular scale, the Saaty's one (SS), contains only 17 numbers - integers 1,2,...,9 and their reciprocals. Other scales suggested in literature are the Extended Saaty's scale (ESS(N)) that contains integers from 1 to N, along with their reciprocals, and the geometric scale GS(c) that contains numbers s of the form $s = c^{i/2}$, $i \in I$ with I being a predefined set of integers.

Obviously there are also other kinds of errors in the DM's judgments that result from the human brain limitations. Consequently, in reality, very rarely the PCM produced by the

DM is the consistent one. Because serious errors in DM's judgments can result in a misleading PCM, an important issue connected with this methodology is the measurement of the degree of inconsistency of the PCM. To deal with this problem a number of inconsistency indices are presented in literature. Two most popular ones are the following.

First, the index proposed by Saaty, denoted as SI , that is defined as follows:

$$SI(n) = \frac{\lambda_{\max} - n}{n - 1}$$

The second one, GI , is related to the geometric mean method and was proposed by Crawford and Williams [4], and popularized by Aguaron and Moreno-Jimenez [1]. It is given by

$$GI(n) = \frac{2}{(n-1)(n-2)} \sum_{i < j} \log^2(a_{ij}w_i/w_j)$$

Apart from these two indices there is also third popular one that is not connected with any prioritization technique. This index is based on the notion of the triad inconsistency. It was proposed by Koczkodaj [12]. Following his idea, for any different $i, j, k \leq n$, a tuple (a_{ik}, a_{ij}, a_{kj}) will be called a *triad*. Koczkodaj proposed to characterize the triad's inconsistency by the number:

$$TI(a_{ik}, a_{kj}, a_{ij}) = \min \left[\left| 1 - \frac{a_{ik}a_{kj}}{a_{ij}} \right|, \left| 1 - \frac{a_{ij}}{(a_{ik}a_{kj})} \right| \right] \quad (1)$$

Next, the Koczkodaj's index KI of any reciprocal PCM is defined as a maximum of triad's inconsistencies i.e. $KI = \max[TI(a_{ik}, a_{kj}, a_{ij})]$, where the maximum is taken over all triads in the upper triangle of the PCM.

III. PROBLEM STATEMENT

All inconsistency indices proposed in literature have only two common features: they are nonnegative and they equal 0 if and only if a given PCM is consistent. But at this point one can notice, that we have infinitely many nonnegative functions of the PCM, that satisfy such conditions. So, we have possibly infinitely many candidates for the inconsistency indices.

In this paper, we study the performance of a new class of inconsistency indices that is based on the notion of a triad inconsistency. The class is defined as follows. Let A be any given PCM, and let T be any real nonnegative function of its triads (a, b, c) such that it equals 0 if and only if $ac = b$. Let $Up(T, A)$ denote a set of values $T(a, b, c)$ obtained for all triads in the upper triangle of A . As inconsistency index II can be taken any function of A defined according the following formula

$$II = Q(Up(T, A)) \quad (2)$$

where the real function Q is linear equivariant, i.e. $Q(aUp(T, A)) = aQ(Up(T, A))$ for any number $a > 0$. Here $aUp(T, A)$ denotes a set of values $\{au : u \in Up(T, A)\}$. Such inconsistency indices are nonnegative and they equal 0 if and only if a given PCM is a consistent one. Certain special cases of indices belonging to the class (2) are known from literature.

E.g. assuming $Q(\cdot) = \max(\cdot)$ and T given by (1) we obtain Koczkodaj's KI , or assuming $Q(\cdot) = \text{Mean}(\cdot)$ and T given by (1) we obtain the index ATI defined in [6]. In [10] an index $MLTI$ is introduced that can be defined by $Q(\cdot) = \text{Mean}(\cdot)$ and $T(a, b, c) = |\ln(ac/b)|$

In our studies we have analyzed many possible indices of the form (2). Because of the limited article space we choose for presentation and discussion only some of them. Apart from the already introduced indices, the most promising ones are the following:

- Index denoted as $ATI2$

$$\text{Mean} \left\{ \left| \frac{a_{ik} - a_{ij}a_{jk}}{a_{ik} + a_{ij}a_{jk}} \right| : 1 \leq i < j < k \leq n \right\} \quad (3)$$

- Indeks denoted as MTI

$$\text{Median} \left\{ \left| \frac{a_{ik} - a_{ij}a_{jk}}{a_{ik} + a_{ij}a_{jk}} \right| : 1 \leq i < j < k \leq n \right\} \quad (4)$$

- Indeks denoted as $ATI3$

$$\text{Mean} \left\{ \left(\frac{a_{ik} - a_{ij}a_{jk}}{a_{ik} + a_{ij}a_{jk}} \right)^2 : 1 \leq i < j < k \leq n \right\} \quad (5)$$

To compare the performance of various considered indices it was proposed in [6] to take into account their relationship with the magnitude of priority-vector-estimation-error (PVEE). This relationship can be examined with the help of Monte Carlo simulations. In [6] it was proposed to measure the PVEE as the average absolute (AE) and/or relative (RE) errors. The errors are defined by the following formulae:

$$AE(\mathbf{v}, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^n |v_i - w_i| \quad (6)$$

$$RE(\mathbf{v}, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^n \frac{|v_i - w_i|}{v_i} \quad (7)$$

where $\mathbf{v} = (v_1, \dots, v_n)$ is the true PV while $\mathbf{w} = (w_1, \dots, w_n)$ is its EPV. Naturally, both the EPV and related errors depend on the prioritization method. In our studies both the GM and REV were used for PV estimation and then the calculation of AE and RE . However, the results presented in this paper are related to the GM.

During the analysis of the multi-criteria-decision-making problem under the AHP scheme, both the prioritization methods and the inconsistency analysis are carried out several times, for each problem-factor separately. Let $\text{PCM}(\text{Cr})$ denote the PCM that was provided by the decision-maker for the criteria, and let the symbol $\text{PCM}(i)$ denote the PCMs achieved for the decision alternatives with respect to the i -th criterion. Let \mathbf{v}^0 and \mathbf{w}^0 be, respectively, the true PV for criteria and the EPV computed for the criteria on the basis of $\text{PCM}(\text{Cr})$. Analogously, let \mathbf{v}^i and \mathbf{w}^i be the true PV and its EPV for the alternatives with respect to the i -th criterion. The true final PV - say \mathbf{v} - is given by

$$\mathbf{v} = \sum_{i=1}^k v_i^0 \mathbf{v}^i \quad (8)$$

The final EPV - say \mathbf{w} - is given analogously:

$$\mathbf{w} = \sum_{i=1}^k w_i^0 \mathbf{w}^i \quad (9)$$

Our principal aim is to analyze the relationship between the values of the considered inconsistency indices and the magnitude of the errors AE , RE computed for the final EPV according the equations (6,7), respectively.

IV. SIMULATION FRAMEWORKS AND RESULTS

To analyze the relationship between the inconsistency indices' values and the PV estimation errors we use a simulation framework that is a modification of the one adopted in [9]. Under this framework a number of "true" PVs are generated, then the corresponding "true" PCMs are randomly disturbed. On the basis of these disturbed PCMs, the related EPVs are computed with the help of the GM. Then the estimation errors (both AE and RE), as well as all necessary problem-characteristics, are stored and finally written down as one database. The steps of our framework are described below in more details.

- Step 0** Set: n - the number of alternatives, k - the number of criteria, N - the number of simulated AHP problems, PD - the probability distribution of PCM-estimation errors
- Step 1** Randomly generate the "true" priority vectors \mathbf{v}^i , $i = 0, \dots, k$,
- Step 2** Compute related "true" comparison matrix \mathbf{M}^i with elements $m_{j,l}^i = \frac{v_j^i}{v_l^i}$
- Step 3** Disturb the elements of $\text{PCM}(\mathbf{v}^i)$, $i = 0, \dots, k$ according to the probability distribution PD .
- Step 4** Compute rounded matrices \mathbf{RM}^i , $i = 0, \dots, k$, by replacing the disturbed elements from the upper-triangle of \mathbf{M}^i , $i = 0, \dots, k$, with the closest values from the adopted scale.
- Step 5** On the basis of the rounded $\text{PCM}(\mathbf{v}^i)$, compute the "estimated" priority vectors \mathbf{w}^i , $i = 0, \dots, k$, with the help of the GM.
- Step 6** Compute the final "true" and "estimated" PVs: \mathbf{v} and \mathbf{w} , respectively
- Step 7** According the formulae (6) and (7) compute the final estimation errors $AE(\mathbf{v}, \mathbf{w})$ and $RE(\mathbf{v}, \mathbf{w})$.
- Step 8** Compute considered-inconsistency-indices-values for all rounded PCMs generated in step 4. Compute the arithmetic mean of these values.
- Step 9** Write down all values computed and/or set in Steps 7 and 8 as one record.
- Step 10** N times repeat Steps 1 to 9
- Step 11** Return all records organized as one database.

Simulation experiments conducted under the above framework enable us to study the relationship between the inconsistency indices values and the quality of the prioritization results reflected by the priority estimation errors. In our experiments, we assume the number of criteria $k = 3, \dots, 6$. In Step 1 the coefficients of the true PVs are generated

independently according to the uniform distribution on the interval $(0, 1)$ (note that the length of the interval does not have any impact on the results, as the PVs are normalized to unity). In Step 3 the probability distributions PD of the multiplicative perturbation factor (i.e. the one that disturbs the elements of PCMs) involve log-normal, gamma, uniform and truncated normal ones. Such distributions of the disturbances are often considered in literature, see e.g. [3], [13] or for more detailed discussion [5], [6]. The parameters of the distributions are set in such a way that their expected values are equal to 1. The support of the truncated normal and the uniform distribution is the interval $DS=[0.5, 1.5]$. The parameters of two remaining distributions are prescribed in a way ensuring that the probability of the interval DS is close to 1. The database generated in our experiments contains 80000 records related to such a number of different AHP setups. To investigate the relationship we are interested in, the whole database is arranged in ascending order with respect to the values of a considered inconsistency index and then split into a given number (say NC) of separate classes - let us denote them as $IC_i, (i = 1, \dots, NC)$. For each such a class the mean value of the considered index is computed as well as the mean value of related errors (AE or RE). Tables I and II present averages (with respect to number of criteria k) of the results obtained for problems where $n = 3, \dots, 9$. In each case the assumed number of classes is $NC = 35$.

Table I contains the Spearman correlation coefficients between the prioritization errors and inconsistency indices - note that only fractional digits are presented. From the perspective of the inconsistency analysis, Spearman correlation coefficient is the most important characteristic of the investigated relationship. That is because any good indicator of the trustworthiness of the PCM at hand should be monotonically related to the prioritization errors. From this perspective, in view of the presented results, the ATI seems to be the best inconsistency index, but also those other ones that are based on the triad inconsistency notion are very competitive. They outperform the most popular indices, that are rather poor with this respect.

Table II contains the Pearson correlation coefficients - they are a bit less important from our point of view because they indicate the "degree of linearity" rather than the monotonicity in examined relationships. However, if additionally, the relationship was linear it would be a nice feature that could enable us to develop the simple regression models that would relate the dependence between given index values and the magnitudes of considered type of error. And indeed, the Pearson correlation coefficients are very high for all new indices, again the ATI has got the highest values. It confirms that it can be easily used in order to predict the magnitude of errors given any of its observed value.

V. FINAL REMARKS

There are two main purposes of the real-world applications of the AHP. The first and perhaps still the most common one is to create rankings in order to point out the best alternative (e.g. the best location for a new factory, the best car to buy or

TABLE I: Spearman correlation coefficients between the prioritization errors and inconsistency indices. Only fractional digits are presented.

Values for the AE								
n :	SI	GI	KI	ATI	ATI2	ATI3	MTI	MLTI
3	863	866	996	997	992	965	992	968
4	930	943	996	996	996	987	995	987
5	975	956	997	998	997	988	998	992
6	972	969	995	997	998	989	999	996
7	979	923	997	1.000	997	989	997	988
8	951	936	994	999	1.0	993	998	993
9	975	958	997	999	998	991	996	998
Values for the RE								
n :	SI	GI	KI	ATI	ATI2	ATI3	MTI	MLTI
3	833	850	995	996	989	960	991	967
4	935	942	995	995	995	988	996	985
5	968	947	997	997	997	989	997	987
6	973	967	995	997	999	989	998	997
7	971	928	997	999	997	988	998	992
8	942	955	994	999	999	994	998	991
9	972	940	995	999	998	994	996	996

the best firm to cooperate with). Another purpose is to allocate certain resources between several individuals/entities (e.g. to allocate limited funds between different scientific institutions or between a number of specific social tasks). In the second case the parts of resources granted to particular entities are proportional to their estimated weights. In such problems the accuracy of the estimates is of primary importance. As we know, this accuracy depends on the trustworthiness of the tuple of PCMs that are produced by the DM. Thus it is important to use such inconsistency index that is strongly monotonically related to the magnitude of the estimation errors. Such an index would allow DMs to distinguish between useful and useless PCMs, and according to their opinion on the acceptable error-size, accept or not the PCMs at hand. In this paper, we introduced a whole class of new inconsistency indices that are based on Koczkodaj's idea of a triad and its consistency. During our research, we investigated a broad range of indices belonging to that class with the goal of identifying the most promising ones. In light of our studies, many of them turned out to be very useful for PCMs-acceptance procedure, and the best among them is the ATI. The ATI is also much better than the still most popular ones: SI and GI . Moreover, our results show that it is possible to develop a trustworthy regression model relating the ATI with the magnitude of estimation errors - it would be a very convenient decision tool. The development of such a model and verification of its usefulness is our next research task.

REFERENCES

- [1] Aguaron, J., Moreno-Jimenez J.M., (2003). "The geometric consistency index: Approximated thresholds." Euro. J. Oper. Res. 147, 137-145.
- [2] Alonso, J., Lamata, T., (2006), "Consistency in the Analytic Hierarchy Process: a New Approach." Inter. J.Uncertain., Fuzzin. Knowl.-Based Syst., 14, 445-459.
- [3] Choo, E.U., Wedley, W.C. (2004), A common framework for deriving preference values from pairwise comparison matrices, Comp. Oper. Res. 31, 893-908.
- [4] Crawford, G., Williams, C.A. (1985), "A note on the analysis of subjective judgment matrices". J. Math. Psychol. 29, 387-405.

TABLE II: Pearson correlation coefficients between the prioritization errors and inconsistency indices. Only fractional digits are presented.

Values for the AE								
<i>n</i> :	SI	GI	KI	ATI	ATI2	ATI3	MTI	MLTI
3	770	767	994	994	978	900	978	933
4	812	825	990	986	973	922	970	944
5	872	869	989	991	981	937	981	959
6	885	887	983	993	986	946	987	969
7	890	875	976	994	985	938	983	963
8	889	873	969	993	985	940	983	964
9	884	868	969	992	983	938	979	963
Values for the RE								
<i>n</i> :	SI	GI	KI	ATI	ATI2	ATI3	MTI	MLTI
3	763	763	994	993	977	898	978	930
4	813	828	990	986	974	923	971	944
5	864	862	989	991	980	934	981	957
6	884	885	983	993	985	944	986	968
7	888	867	975	993	984	934	982	961
8	882	870	968	992	983	936	981	962
9	883	866	967	991	981	936	978	961

- [5] Grzybowski, A. Z., (2012), "Note on a new optimization based approach for estimating priority weights and related consistency index", *Expert Systems with Applications* 39, 11699-11708.
- [6] Grzybowski, A.Z.,(2016), "New results on inconsistency indices and their relationship with the quality of priority vector estimation", *Expert Systems With Applications* 43, 197-212.
- [7] Grzybowski A. Z., Starczewski T., "Remarks about Inconsistency Analysis in the Pairwise Comparison Technique", *Proceedings of IEEE 14th International Scientific Conference on Informatics, Poprad*, (eds. Novitzka, S. Korecko, A. Szakal), New York 2017, pp.227-231.
- [8] Grzybowski A. Z., Starczewski T., "Simulation analysis of prioritization errors in the AHP and their relationship with an adopted judgement scale", *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2018*, 23-25 October, 2018, San Francisco, USA, pp. 47-51.
- [9] Grzybowski A. Z., Starczewski T.,(2019),"Analysis of Errors in Priority Vector Estimation and Their Relationship with the Correctness of the Final Ranking of Decision Alternatives," *IAENG International Journal of Applied Mathematics*, vol. 49, no. 3, pp. 381-386.
- [10] Kazibudzki, P.T.,(2016), "Redefinition of triad's inconsistency and its impact on the consistency measurement of pairwise comparison matrix", *Journal of Applied Mathematics and Computational Mechanics*, vol. 15, no. 1, pp. 71-78.
- [11] Kazibudzki P.T. Grzybowski, A.Z.,(2013), "On some advancements within certain multicriteria decision making support methodology", *American Journal of Business and Management*, Vol. 2, No. 2, pp. 143-154.
- [12] Koczkodaj W.W. (1993) "A new definition of consistency of pairwise comparisons." *Mathematical and Computer Modelling* 18(7),79-84.
- [13] Lin, C-C. (2007) "A Revised Framework for Deriving Preference Values from Pairwise Comparison Matrices". *Euro. J. Oper. Res.*, 176, 1145-1150.
- [14] Saaty, T.L. (1980) *The Analytic Hierarchy Process*, McGraw Hill, New York.
- [15] Saaty, T.L., Vargas, L.G. (1984) "Comparison of eigenvalue, logarithmic least square and least square methods in estimating ratio". *J. Math. Model.*, 5, 309-324.
- [16] Saaty, T.L., Vargas L.G.(2013) *The Logic of Priorities: Applications of the AHP in Business, Energy, Health and Transportation*, RWS Publications, Pittsburgh (ebook).
- [17] Zyoud S. H., Fuchs-Hanusch D. (2017). "A bibliometric-based survey on AHP and TOPSIS techniques", *Expert Systems With Applications*, 78, 158-181.

Towards More Sophisticated Static Analysis Methods of Python Programs

Hristina Gulabovska

Department of Programming Languages and Compilers
Eötvös Loránd University
 Budapest, Hungary
 hristina@gulab.me

Zoltán Porkoláb

Department of Programming Languages and Compilers
Eötvös Loránd University
 Budapest, Hungary
 gsd@elte.hu

Abstract—Static analysis is a software verification method which is analyzing the source code without executing it for detecting code smells and possible software bugs. Various analysis methods have been successfully applied for languages with static type system, such as C, C++ and Java. Python is an important programming language with dynamic type system, used in many emerging areas, including data science, machine learning and web applications. The dynamic behavior of the Python language requires different static analysis approaches compared to the ones with static type system. In this paper we overview these methods and investigate their advantages and shortages. We compare the symbolic execution with the generally used Abstract Syntax Tree based approach and show its advantages based on concrete examples. We also highlight the restrictions of current tools and suggest further research directions to tackle these problems.

Index Terms—static analysis, symbolic execution, Python

I. INTRODUCTION

Python is one of the most rapidly emerging programming language [1]. Being flexible and expressive, it is very popular to implement Machine Learning and Cloud based systems among others. The popularity is partially derived from its dynamic behavior: Python is a dynamically typed programming language, i.e. a variable is just a value bind to a certain (variable) name, the value has a type but not the variable. One can assign to an existing variable a new value with a possibly different type. This would be a compile time error in statically typed languages such as Pascal, C or Java, but allowed in Python. In the same time, Python is strongly typed, as operations may fail on an object when the operation is not defined on the actual type of the value hold [2]. More dynamic features, such as calling methods dynamically, declaring dynamic attributes (using `getattr`), and others also increase the expressiveness and usability of the language.

However, such dynamic behavior might be an obstacle when we try to validate the software systems written in Python. For languages with static type system various static analysis methods exist and used either as commercial [3], [4] or free, open source tools [5]–[8]. Although there are promising projects for Python (see Section III), they are significantly less expressive than their counterparts for C, C++ or Java languages.

In the recent years we experienced rapid development of static analysis tools and methods. Besides proprietary tools

we found a fine number of open source projects with growing developers’ communities [9], [10]. Static analysis is an important aspects in modern software development, addressed by various academic and industrial researches, and projects like by the Intellectual outputs No. O1 and O2 of the Erasmus+ Key Action 2 (Strategic partnership for higher education) project No.2017-1-SK01-KA203-035402: “Focusing Education on Composability, Comprehensibility and Correctness of Working Software” [11], [12].

In this paper we investigate the possible research directions towards more powerful static analysis tools for the Python programming language. In Section II we evaluate the applicable analysis methods evaluating their strength and weaknesses. In Section III we briefly overview the most important tools and research directions currently available for analyzing Python systems. We use two specific tools to compare the AST based methods and the symbolic/concolic execution in Section IV. Our paper concludes in Section V.

II. STATIC ANALYSIS METHODS FOR SOFTWARE SYSTEMS

To verify the correctness of a software system we can choose between various methods. The most common solution is to write test cases, either using white box or black box approach. Although testing is essential for modern software development, it is a costly and slow approach which efficiency greatly depends on the test coverage we achieve. As the earlier a bug is detected, the lower the cost of the fix [13], testing is not ideal in this aspect.

Alternatively we might turn to analyzer tools which applies various validation methods to find potential or actual misuses in the software. Dynamic analysis tools run the program in a special environment where they can detect incorrect, erroneous execution. However, tools such as *Valgrind* [14], or *Google Address sanitizer* [15] which work run-time, evaluate the correctness of only those parts of the system which have been actually executed. Such *dynamic analysis methods* therefore require carefully selected input data, and they easily can miss certain corner cases.

In the case of static analysis we do not run the software. Instead, the input of the static analysers are the source code, and we apply various methods to find dangerous constructs without running the program. Static analysis is a popular

method for finding bugs and code smells [16] as they do not depend on the selection of input data while they can (at least theoretically) provide full coverage of the code. An other advantage of the static analysis method is that it is many cases applicable for only a part of the code. This is useful when we have no full control over the system, e.g. we use third party libraries, not all source is available, or we just have no resources to check the whole system.

Most static methods apply heuristics, which means that sometimes they may *underestimate* or *overestimate* the program behavior. In practice this means static analysis tools sometimes do not report existing issues which situation is called as *false negative*, and sometimes they report correct code erroneously as a problem, which is called as *false positive*. Therefore, all reports need to be reviewed by a professional who has to decide whether the report stands.

Here we face an interesting human challenge. Unlike the theoretical program verification methods, we do not only strive to minimize the false negatives, i.e. try to find all possible issues, but we aspire to *minimize the false positives*. If the tool produces a large number of false positives, the necessary review process takes unreasonable large effort by the developers, meanwhile they also lose their trust in the analyser tool. A general approach for static analysis tools to try the balance between the large number of true positives and a relatively low number of false positives. When in doubt, many tools rather choose to drop real findings to minimize false alarms.

A. Pattern matching

In this method the source code is first converted to a canonical format, and we match regular expression to every line in the source and reports each matches. Although, this method seems to be very simple, its huge advantage is the low level of false positives, as well-written regular expressions have easy to predict results. Early versions of CppCheck [17] used pattern matching to find issues in C and C++ programs. Additional advantage of pattern matching is working on non-complete source, even if we cannot recompile the software system. This means that the method can be applied to software under construction, or when we have only partial information on the system.

In the same time this method has several disadvantages too. As regular expressions are context free grammars, we are restricted to find issues based on information in the close locality of the problem. Having no knowledge about declarations we could not use type information. We cannot use name and overload resolution, cannot follow function calls and generally we have weak understanding on the overall program structure. As a summary, we can consider pattern matching based approaches as easy entry-level methods [18].

B. AST matchers

To provide the necessary context information to the analysis we can use the *Abstract Syntax Tree* (AST). AST is the usual internal data structure used by the front-end phase of the

compilers [19]. Basically, the AST is a lossless representation of the program; it encodes the structure of the program, the declarations, the variable usages, function calls, etc. Frequently, the AST is also decorated with type information and contains connections between declarations (types, functions, variables) and their usage.

This representation is suitable for catching errors that the simple pattern matching is unable to detect. These issues include heap allocation using the wrong size, some erroneous implicit conversions, inconsistent design rules, such as hurting the rule of three/five in C++ and it proved to be strong enough to detect even some misuses of the STL API.

Such AST based checks are usually relatively fast. Some rules can even be implemented using a single traversal of the AST. That makes it possible to implement such checks as editor plug-ins. External tools, such as the Clang Tidy [6] uses AST matching for most of its checks.

While the AST matcher method is more powerful than the simple pattern matching, it has some disadvantages too. To build up a valid AST requires a complete, syntactically correct source file. To resolve external module dependences we need some additional information not represented in the source itself, such as include path for C/C++ programs, CLASSPATH for Java or BASE_DIR in Python. That usually means, we have to integrate the static analysis tool into the build system which can be painful. Another shortage of the AST matcher method, that it cannot reason about the possible program states which can be dependant from input values, function parameters.

C. Symbolic execution

Executing *abstract interpretation* [20] the tool reasons about the possible values of variables at a certain program point. *Symbolic execution* [21], [22] is a path-sensitive abstract interpretation method. During symbolic execution we interpret the source code, but instead of using the exact (unknown) run-time values of the variables we use symbolic values and gradually build up constraints on their possible values. Memory locations and their connections (e.g. a pointer pointing to some variable, structs and their fields) are represented by a sophisticated hierarchical memory model [23]. A constraint solver can reason about these values and is used to exclude unviable execution paths. Most of the high end proprietary analysis tools, such as CodeSonar [7], Klocwork [4], and Coverity [3] as well as open source products such as the Clang Static Analyzer [5], and Infer [8] use this method.

Symbolic execution is the most powerful method for static analysis as it makes profit from the program structure, the type system, the data flow information and is able to follow function calls. However, there is a price for this. To represent the internal state of the analysis, the analyzer uses a data structure called the *exploded graph* [24]. Each vertex of this graph is a (symbolic state, program point) pair. The edges of the graph are transitions between vertices. This graph is exponential in the number of control branches (conditions) of the program. This could be critical, especially for loops, which are represented as unrolled set of conditions and statements.

This factor makes symbolic execution also the most resource expensive (in memory and execution time) method.

D. Concolic execution

While the symbolic execution of a program can generate all possible control flow path that can occur during the execution, in practice it is unfeasible for many reasons. Most of the constraint solvers has some limitations on their modelling capabilities. Also, generic constraint solvers working e.g. on intervals may be resource (time and memory) intensive thus make full symbolic execution impossible.

An interesting mixture of symbolic and concrete execution is called as *concolic* execution [25] targets this problem. The main idea is that we use concrete values for execution driven by symbolic execution. We start the execution with an arbitrary input value, we maintain both the symbolic execution state and a storage for the concrete values. Whenever the concrete execution takes a branch, the symbolic execution is directed toward the same branch. Then the constraint solver is used to negate the path conditions thus to choose a new concrete value to cover the other branch. [26].

The advantage of concolic execution is that the operations of the program state can be executed on concrete values, thus it could be implemented in more simple way and using less resources, while the SAT solver still helps to keep follow all the possible execution paths. Especially for languages such as Python, where the interpreter could evaluate the analysed program making possible to call unmodeled external methods or using third party modules this approach is seriously extending the power of the symbolic execution.

III. ANALYSIS TOOLS FOR PYTHON

Compiler relies on static analysis to generate its warnings during compilation time. However, since its primary task is to translate source code from a high-level programming language to a lower level language, it is less capable of doing static analysis than some of the third-party tools. The Python compiler misses catching a number of common bugs and errors, therefore already existing third-party Python static analysis tools are aiming to cover the catches missed by the actual compiler.

Among the most common actual Python static analysis tools, the following could be listed as the most reliable: Pylint [27], Pyflakes [28], flake8 [29], Frosted [30], Pycodestyle [31], and Mypy [32]. These tools are open-sourced, and some of them are still explicitly said to be in an experimental stage. They are using the AST method in order to statically evaluate the potential bug and errors of the source code. Pylint is seen as the most popular Python static analysis tool at the moment, which is free and capable of not only catching logical errors but also warns regarding the specific coding standards. In Pylint, there is a possibility to write custom rules too. There are three types of possible custom rules: Raw checkers (analyzing each module as a raw file stream), Token checkers (using list of tokens representing the source code) and most of the checkers are working on

the Abstract Syntax Tree (AST) which is provided by the *astroid* [33] library. Adding to the reliability of Pylint, it is worth mentioning that it is trusted by many big companies, such as Google [34], which is mostly relying on Pylint for the static analysis of its Python code-base. There is also a number of popular IDEs and frameworks using Pylint for in-time static analysis of the Python code, some of which: PyCharm, VSCode, Django with Pylint, Eclipse with PyDev etc.

Beside the present static analysis tools, there are several Python tools (mostly in experimental status) which are related to symbolic execution and SMT (Satisfiability Modulo Theories) such as: PyExZ3 [35], PySym [36], PySMT [37], and mini-mc [38], etc. Most of them are using the Z3-solver [39].

During the research and comparison of the AST and symbolic execution methods for Python static analysis in this paper, two tools were used. Pylint, as the currently most reliable representation of evaluating the AST, and for the symbolic execution part, Z3-solver and mini-mc symbolic model checker, which helps to explore the symbolic evaluation of the source code.

One of the more critical common bugs in Python that was not caught by Pylint during the research, nor the Python compiler itself, was the “Closure bug”. Closure in Python is an important concept that allows the function object to remember the values in enclosing scopes even if they are not present in memory. At the same time it is prone to bugs which as shown in the code example on Listing 1 is very often hardly caught even during runtime.

```

1 def greet(greet_word, name):
2     print(greet_word, name)
3
4 greeters = list()
5 names = ["Kiki", "Riki", "Joe"]
6 for name in names:
7     greeters.append(lambda x: greet(x, name))
8 for greeter in greeters:
9     greeter("Hi ")

```

Listing 1: The closure bug

We may expect that this code would printout:

```

1 Hi Kiki
2 Hi Riki
3 Hi Joe

```

But instead it prints:

```

1 Hi Joe
2 Hi Joe
3 Hi Joe

```

The closure bug is one of the most tricky issue without actually causing a run time error. In our earlier researches we found that only Pylint is able to catch this problem, reporting a Warning “Cell variable name defined in loop” which is not necessary a clear message for the developers about the specific error they made.

IV. EVALUATION

In this section, we compare the power of the AST-based method with the symbolic execution. We were selecting two representative tools for the two methods, run tests with them, and analyze the results. Our goal is not only to show which methods can report more real errors, *true positives*, but also which are better avoiding to report *false positives* – code snippets that are correct but falsely reported a suspicious code segment.

Most of the currently existing Python tools use the AST method for static analysis of the Python source code. We have selected Pylint [27] as one of the most widely used and most powerful static analysis tool for Python which is using the AST method based on the asteroid library.

For the symbolic execution method we have chosen mini-mc, an experimental symbolic execution implementation [38], using the Z3's Python interface. The mini-mc tool is implementing the fork-explore-check idea, i.e. when evaluating symbolic values the Python VM will try to convert them into boolean values at all branches to intercept the conversion and replace it with a `fork` statement. In practice mini-mc will process all reachable program paths, forking new process to evaluate the false branch. It also detects unreachable conditions where the evaluation stops. We have chosen mini-mc for its simplicity and demonstrative power.

Analyzing the methods for static analysis and noticing that symbolic execution might be the best approach for static analyzing of Python considering its dynamically typed characteristics. In this section, we composed a few examples to see step by step the symbolic evaluation and then compare if Pylint as a AST based static analyzer or mini-mc as a symbolic execution method could catch the errors during the analysis and exclude false positives in unreachable paths.

```

1 #!/usr/bin/env python3
2 from mc import *
3 import os
4 import time
5
6 def func(arg):
7     if(1==arg):
8         print("branch11",os.getpid())
9         x=1
10    else:
11        print("branch12",os.getpid())
12        x=0
13    if(1==arg):
14        print("branch21",os.getpid())
15        y=5/x
16    else:
17        time.sleep(3)
18        print("branch22",os.getpid())
19        y=4/(x+1)
20 arg = BitVec("arg",32)
21 func(arg)

```

Listing 2: Usage example of mini-mc.

Listing 2 is the very first Python code example that we used to run the symbolic model checker, and as it is seen, this

program should not report an error since both if-conditions (in line 7 and line 13) could be true at the same time and their bodies could be executed without errors. We used this example mostly to demonstrate the way of execution of the mini-mc symbolic model checker. As it is seen on Listing 3 the program was executed in a quasi parallel way. At every branch statement the program forks a new process, the process id and the logical assumption is written to the output. (The use of `time.sleep(3)` on line 17 is to show the non deterministic evaluation order of the branches). When the engine detects unsatisfiable condition, that is also printed as `unreachable`.

```

1 [7088] assume (arg == 1)
2 [7090] assume ¬(arg == 1)
3 [7090] unreachable
4 [7088] assume (arg == 1)
5 [7088] assume (arg == 1)
6 branch11 7088
7 branch11 7088
8 branch21 7088
9 [7090] exit
10 [7089] assume ¬(arg == 1)
11 [7089] assume (arg == 1)
12 [7089] unreachable
13 branch12 7089
14 [7089] assume ¬(arg == 1)
15 [7091] assume ¬(arg == 1)
16 branch12 7089
17 branch22 7091
18 [7091] exit
19 [7089] exit
20 [7088] exit

```

Listing 3: Execution result of Listing 2.

The next Python example on Listing 4 is faulty and should point out the power of symbolic execution over the AST based approaches. As the two if-conditions could not be true at the same time, if `1!=arg` then the first if-condition body would not be executed, and when the second if-condition body will be executed, the program gets faulty since the variable `z` was not introduced yet.

```

1 def func(arg):
2     if(1==arg):
3         print("branch11",os.getpid())
4         z=1
5     if(1!=arg):
6         print("branch21",os.getpid())
7         x=z
8
9 arg = BitVec("arg",32)
10 func(arg)

```

Listing 4: Local variable referenced before assignment.

The result shows that symbolic execution finds the possible error while the AST based approach will miss it. When Pylint as an AST based static analyzer was run, it did not detect and report any potential error in the program. Pylint was unable to recognize that `x=z` will be executed only after statement `z=1` is not. Pylint is assuming that variable `z` is defined in one of the executed branches and conservatively do not report error to minimize possible false positives.

When the mini-mc symbolic execution checker was run and the steps of the execution were provided it could be seen that the Unbound error at line 14 was detected (Listing 5). This shows the benefits of the symbolic execution over the AST based methods for Python as a dynamic language.

```

1 [6324] assume (arg == 1)
2 [6324] assume (arg != 1)
3 [6324] unreachable
4 branch11 6324
5 [6324] assume (arg == 1)
6 [6326] assume ¬(arg != 1)
7 branch11 6324
8 [6326] exit
9 [6325] assume ¬(arg == 1)
10 [6327] assume ¬(arg != 1)
11 [6327] unreachable
12 [6327] exit
13 [6325] assume ¬(arg == 1)
14 [6325] assume (arg != 1)
15 branch21 6325
16 Traceback (most recent call last):
17   File "./example4.py", line 17, in <module>
18     func(arg)
19   File "./example4.py", line 14, in func
20     x=z
21 UnboundLocalError: local variable 'z' referenced
    before assignment
22 [6325] exit
23 [6324] exit

```

Listing 5: Execution result of Listing 4.

On Listing 6 we changed the code in order to check the behavior of symbolic evaluation when instead of concrete values, intervals are used in the if conditions. In order to enter the second if-condition one has to also enter the first if-condition and the unbounded error should not be reported. The results on Listing 7 shows that the symbolic execution was working just fine when we used intervals.

```

1 def func(arg):
2     if(1<arg):
3         print("branch11", os.getpid())
4         z=1
5     if(2<arg):
6         print("branch21", os.getpid())
7         x=z

```

Listing 6: Using intervals in conditions.

```

1 [5243] assume (arg > 1)
2 [5243] assume (arg > 2)
3 [5243] assume (arg > 1)
4 [5245] assume ¬(arg > 2)
5 branch11 5243
6 branch21 5243
7 branch11 5243
8 [5245] exit
9 [5244] assume ¬(arg > 1)
10 [5244] assume (arg > 2)
11 [5244] unreachable
12 [5244] assume ¬(arg > 1)
13 [5246] assume ¬(arg > 2)
14 [5246] exit
15 [5244] exit
16 [5243] exit

```

Listing 7: Execution result of Listing 6.

There are certain situations, however, when symbolic/concolic execution may cause unreasonable false positives. This is derived from the nature of concolic execution we discussed in Section II-D. The evaluation is partially driven by the SAT solver, that is the engine to encounter all execution paths, but the concrete execution of the statements inside the branches is using a concrete value chosen by the constraint.

In the example on Listing 8 we execute a function with an unknown argument `arg`. There is a very small possibility, that `arg` is 42, which could cause `ZeroDivisionError`. PyLint static analyzer does not report such an error, as this would be most likely a false positive.

```

1 def func(arg):
2     if arg == 41:
3         print("branch21", os.getpid())
4     else:
5         print("branch22", os.getpid())
6         z = arg - 42
7         z = 99 / z

```

Listing 8: Concolic execution

The following result on Listing 9 shows that the symbolic execution method inaccurately reports the possibility of `ZeroDivisionError` for the else branch.

```

1 [15532] assume (arg == 41)
2 branch21 15532
3 [15533] assume ¬(arg == 41)
4 branch22 15533
5 42
6 Traceback (most recent call last):
7   File "example11.py", line 23, in <module>
8     func(arg)
9   File "example11.py", line 17, in func
10     z = 99 / z
11 ZeroDivisionError: division by zero
12 [15533] exit
13 [15532] exit

```

Listing 9: False positive report of Listing 8.

The reason is, that the concolic execution accidentally takes 42 as the sample value for `arg` when `arg != 41`. This makes the otherwise unlikely situation of dividing by zero unavoidable. Although such unlucky situations may be un-frequent in every day development, this false positive could be extremely annoying.

Nevertheless, at the moment concolic execution seems to be the most powerful static analysis method for dynamic languages such as Python, this example shows that it is far from perfect and there is room to improve it.

V. CONCLUSION

Static analysis methods evaluate software systems without running them, applying various heuristics on the source code to detect possible code vulnerabilities. Since these do not require to choose specific input values, they are easier to integrate into the Continuous Integration loop. However, static analysis tools provide less support for programming languages with

dynamically type system, such as Python. Although there exist some – mainly AST based – tools, their capacity to detect Python-specific errors are not satisfactory.

Symbolic execution techniques may open new directions for supporting Python static analysis. The dynamic behavior of the language, such as the usage of variables without preceding declaration and changing its type during runtime is better covered by methods, where the change of the program state is emulated. With the help of SAT solvers, we can provide full coverage of the program path (under the resource constraints). We have shown that even the most simple symbolic execution tools can find issues otherwise not detected by AST based tools. Especially concolic execution is one of the most promising method directions.

Symbolic execution based analyzer tools for Python can use powerful SAT solvers, such as Z3, but currently, they model neither the type of information nor the most important modules of the Python language. This a serious restriction, and further research should target that area. Nevertheless, the otherwise powerful concolic execution can also cause unwanted false positives. Based on our experiments, we suggest using both an AST based tool and a symbolic execution tool to maximize the true positives and minimize the reported false positives.

ACKNOWLEDGMENT

This work is supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

REFERENCES

- [1] Tiobe. (2019) TIOBE programming community index, July 2019. TIOBE Software. Accessed 02-July-2019. [Online]. Available: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [2] G. van Rossum, "Python tutorial," Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Tech. Rep. CS-R9526, May 1995.
- [3] Synopsys, "Coverity," 2019, <https://scan.coverity.com/> (last accessed: 28-02-2019).
- [4] Roguewave, "Klocwork," 2019, <https://www.roguewave.com/products-services/klocwork> (last accessed: 28-02-2019).
- [5] Clang SA, "Clang Static Analyzer," 2019, <https://clang-analyzer.lvm.org/>.
- [6] Clang Tidy, "Clang-Tidy," 2019, <https://clang.lvm.org/extra/clang-tidy/> (last accessed: 28-02-2019).
- [7] GrammaTech, "CodeSonar," 2019, <https://www.grammatel.com/products/codesonar> (last accessed: 28-02-2019).
- [8] C. Calcagno and D. Distefano, "Infer: An automatic program verifier for memory safety of C programs," in *NASA Formal Methods Symposium*. Springer, 2011, pp. 459–465.
- [9] A. Dergachev, "Clang Static Analyzer: A Checker Developer's Guide," 2016, <https://github.com/haoNoQ/clang-analyzer-guide> (last accessed: 28-02-2019).
- [10] A. Zaks and J. Rose, "Building a checker in 24 hours," 2012, <https://www.youtube.com/watch?v=kdxlsP5QVPw>.
- [11] Š Korečko, "Interactive approach to coloured petri nets teaching," Eötvös Loránd University, Faculty of Informatics, Budapest, Tech. Rep. IK-TR3, May 2018.
- [12] C. Szabó. (2018) Programme of the winter school of project no.2017-1-sk01-ka203-035402: "focusing education on composability, comprehensibility and correctness of working software". TUKE Kosice. Accessed 02-July-2019. [Online]. Available: https://kpi.fe.i.tuke.sk/sites/www2.kpi.fe.i.tuke.sk/files/personal/programme_of_the_first_intensive_programme_for_higher_education_learners_in_the_frame_of_the_project.pdf
- [13] B. Boehm and V. R. Basili, "Software defect reduction top 10 list," *Computer*, vol. 34, no. 1, pp. 135–137, Jan. 2001. [Online]. Available: <http://dx.doi.org/10.1109/2.962984>
- [14] N. Nethercote and J. Seward, "Valgrind: A framework for heavyweight dynamic binary instrumentation," *SIGPLAN Not.*, vol. 42, no. 6, pp. 89–100, Jun. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1273442.1250746>
- [15] K. Serebryany, D. Bruening, A. Potapenko, and D. Vyukov, "Addresssanitizer: A fast address sanity checker," in *Proceedings of the 2012 USENIX Conference on Annual Technical Conference*, ser. USENIX ATC'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 28–28. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2342821.2342849>
- [16] A. Bessey, K. Block, B. Chelf, A. Chou, B. Fulton, S. Hallem, C. Henri-Gros, A. Kamsky, S. McPeak, and D. Engler, "A few billion lines of code later: Using static analysis to find bugs in the real world," *Commun. ACM*, vol. 53, no. 2, pp. 66–75, Feb. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1646353.1646374>
- [17] D. Marjamäki, "CppCheck: a tool for static C/C++ code analysis," 2013. [Online]. Available: <http://cppcheck.sourceforge.net/>
- [18] M. Moene, "Search with cppcheck," *Overload Journal*, vol. 120, 2014. [Online]. Available: <https://accu.org/index.php/journals/1898>
- [19] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers principles, techniques, and tools*. Reading, MA: Addison-Wesley, 1986.
- [20] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. ACM, 1977, pp. 238–252.
- [21] H. Hampapuram, Y. Yang, and M. Das, "Symbolic path simulation in path-sensitive dataflow analysis," *SIGSOFT Softw. Eng. Notes*, vol. 31, no. 1, pp. 52–58, Sep. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1108768.1108808>
- [22] J. C. King, "Symbolic execution and program testing," *Commun. ACM*, vol. 19, no. 7, pp. 385–394, Jul. 1976. [Online]. Available: <http://doi.acm.org/10.1145/360248.360252>
- [23] Z. Xu, T. Kremenek, and J. Zhang, "A memory model for static analysis of C programs," in *Proceedings of the 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation - Volume Part I*, ser. ISO/LA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 535–548. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1939281.1939332>
- [24] T. Reps, S. Horwitz, and M. Sagiv, "Precise interprocedural dataflow analysis via graph reachability," in *Proceedings of the 22Nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '95. New York, NY, USA: ACM, 1995, pp. 49–61. [Online]. Available: <http://doi.acm.org/10.1145/199448.199462>
- [25] K. Sen, "Concolic testing," in *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '07. New York, NY, USA: ACM, 2007, pp. 571–572. [Online]. Available: <http://doi.acm.org/10.1145/1321631.1321746>
- [26] R. Baldoni, E. Coppa, D. C. D'Elia, C. Demetrescu, and I. Finocchi, "A survey of symbolic execution techniques," *ACM Comput. Surv.*, vol. 51, no. 3, 2018.
- [27] Logilab. (2003) Pylint. <http://pylint.pycqa.org/en/latest/>.
- [28] PyCQA. (2014) Pyflakes. <https://pypi.org/project/pyflakes/>.
- [29] I. Cordasco. (2016) Flake8. <http://flake8.pycqa.org/en/latest/>.
- [30] T. Crosley. (2014) Frosted. <https://pypi.org/project/frosted/>.
- [31] J. Rocholl. (2006) Pycodestyle. <http://pycodestyle.pycqa.org/en/latest/>.
- [32] J. Lehtosalo. (2016) Mypy. <https://mypy.readthedocs.io/en/latest/>.
- [33] Logilab. (2019) Astroid. <https://astroid.readthedocs.io/en/latest/>.
- [34] Google. (2018) Google python style guide. <http://google.github.io/styleguide/pyguide.html>.
- [35] T. Ball and J. Daniel, "Deconstructing dynamic symbolic execution," *Proceedings of the 2014 Marktoberdorf Summer School on Dependable Software Systems Engineering*, no. MSR-TR-2015-95, January 2015. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/deconstructing-dynamic-symbolic-execution/>
- [36] B. I. Dahlgren. (2016) Pysym. <https://pythonhosted.org/pysym/>.
- [37] Y. Malheiros. (2016) pysmt. <https://pysmt.readthedocs.io/en/latest/tutorials.html>.
- [38] X. Wang. (2015) A mini symbolic execution engine. <http://kqueue.org/blog/2015/05/26/mini-mc/>.
- [39] L. de Moura and N. Björner, *Z3: An Efficient SMT Solver*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 4963, pp. 337–340. [Online]. Available: https://doi.org/10.1007/978-3-540-78800-3_24

The Use of Mini Computer in Hybrid FSO/RF System

Renát Haluška
Dept. of Electronics and Multimedia
Communications
Technical University of Košice
Košice, Slovakia
renat.haluska@tuke.sk

Luboš Ovseník
Dept. of Electronics and Multimedia
Communications
Technical University of Košice
Košice, Slovakia
lubos.ovsenik@tuke.sk

Peter Šulaj
Dept. of Electronics and Multimedia
Communications
Technical University of Košice
Košice, Slovakia
peter.sulaj@tuke.sk

Abstract—This article discusses the problem of optical communication with Free Space Optics (FSO), its use, and the conditions that affect it. This document deals with the practical design and construction of the monitoring equipment. Such a device is designed to collect and process data characterizing the nature of the conditions along the transmission path. Because of the vulnerability of the FSO transmission channel to weather conditions, it is necessary to predict the weather and so switch to a backup line using machine learning method called decision tree.

Keywords— FSO, hybrid system, machine learning, mini-PC

I. INTRODUCTION

Free-space optical communication (FSO) means the transmission of optical signals through free space such as air. Such dissemination of data over the air requires the use of an optical signal over the light. The light sources for transmission may be either light-emitting diodes or lasers (amplification of light stimulated by emission of radiation). FSO is a simple concept that is similar to fiber optic transmission. The main difference is in the medium used, and one of the attractions is that the light rays move through the air faster than through glass, which uses fiber transfer at a speed of approximately 200,000 km/s. From this we can conclude that optical communication through free space can be classified as optical communication at light speed [1].

Since the Regulatory Office for Electronic Communications and Postal Services (RU) does not regulate frequency usage above 300 GHz, unlike most low frequency microwave systems, FSO communication systems do not require operating licenses. This is true not only in Slovakia but worldwide. Due to the proximity of the IR spectrum used by FSO to the visible spectrum, it has almost the same properties as visible light.

The emitted light is collected on the receiver side using a lens. Subsequently, the received light is aimed at a photoelectric detector. For all practical purposes, the projected beam size at the receiving end is much larger than the size of the receiving optic. Therefore, part of the transmitted light is lost during the transmission process. Depending on the current beam divergence, the projected beam size may be several meters, while the typical diameter of the receiving telescope is several centimeters. This phenomenon is called the loss of

the geometric path. However, narrow beams require a very stable mounting platform or a more sophisticated active beam tracking system [2].

The main requirement for the operation of fiber-free optical systems is seamless visibility between two communication points, as FSO systems use light for communication and this cannot pass through solid obstacles such as walls, buildings, trees and others.

II. HYBRID FSO/RF SYSTEM

The performance of FSO communications is affected by weather conditions and atmospheric turbulence. This can lead to a loss of connection or slow data transfer. Therefore, in order to increase reliability and improve connection availability, it is appropriate to combine the FSO system with a more reliable radio frequency (RF) system. Such systems are called hybrid FSO/RF systems (Fig. 1). These systems are capable of providing high availability even under adverse weather conditions [3]. The cause of the most degrading signal in RF transmission is rain, because the carrier wavelength is nearly identical to the size of the rain drop, FSO communication is affected by fog. The overall availability of the system can be improved by using a low data rate link as a back-up when the FSO connection cannot be used.

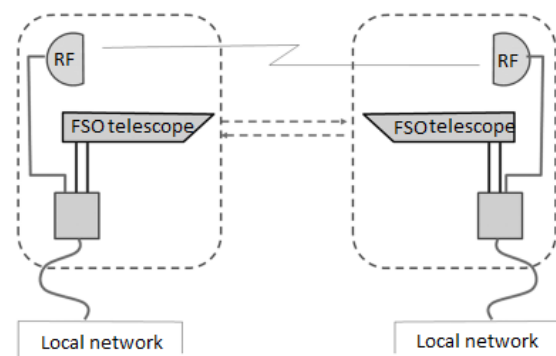


Fig. 1 Hybrid FSO/RF System [4]

Optical beam that spreads through a turbulent atmosphere is subjected to another effect of diffraction diffusion [8]. The most common optic beam is the so-called *Gaussian beam* from the distance L from the radiation source under turbulent

atmospheres. The intensity of the time-averaged optical beam can be expressed according to the following equation

$$I(l, r) = \frac{2P_0}{\pi\omega_{eff}^2(l)} \exp\left[\frac{-2r^2}{\omega_{eff}^2(l)}\right]. \quad (5)$$

The most commonly used qualitative characteristics, not only in the optical communication systems, are the *bit error rate* (BER) and the *Signal/Noise Ratio* (SNR). The BER performance parameter is based on average received optical output, scintillation intensity and, last but not least, the receiver amount [6]. By appropriately accommodating the dimensions of the receiving aperture, it is possible to increase the intensity of the received optical power and thus to negatively effect the scintillation effect. The value of the SNR parameter in the result of the turbulence can be expressed by the following equation

$$SNR = \frac{1}{0.31C_n^2 k^{7/6} l^{11/6}} \quad (6)$$

FSO systems use ON / OFF modulation for which it is possible to express bit error rate BER according to the following equation

$$BER = \frac{\exp(-SNR/2)}{(2\pi SNR)^{0.5}} \quad (7)$$

Assuming that the receiving area of the photo detector of the FSO line head is sufficiently large, it is necessary to calculate the SNR with the influence of the magnification of the optical beam. In this case, the so- effective SNR, the calculation of which is given in the following equation

$$SNR_{eff} = \frac{SNR}{1 + 1.33\sigma_i^2 \left[\frac{2l}{k\omega(l)^2} \right]^{5/6}} \quad (8)$$

where P_0 is the total power of the optical beam W , and r is the radial distance from the center of the beam. The quality of the expanded optical beam is accentuated as a result of the lowest beam waist $\omega_{eff}(l)^2 > \omega_{eff}(l)$ [7]. The quantization of the beam can be expressed by the following equation

$$\omega_{eff}(l)^2 = \omega(l)^2 + (1 + T) \quad (9)$$

where $\omega(l)$ expresses the change of ω after passing the distance L [m] expressed

$$\omega(l)^2 = \left[\omega_0^2 + \left(\frac{2L}{k\omega_0} \right)^2 \right] [m^2] \quad (10)$$

Parameter ω_0 is the initial velocity of the beam extension at a distance ($L = 0m$) from the light source and T is additive beam expansion in the result of turbulence.

The first head of a used FSO line pair located in the campus of the Technical University in Košice (TUKE) is located on the PK13 building (Fig.2). The second FSO head is located on the roof of the TUKE main building L9A (Letná 9). Distance between FSO heads is around 230 meters.



Fig. 2 FSO heads installed at the PK13 building

Each of the FSO headlamps is equipped with two connectors for connecting the optical multi-vision fibers. The first optical fiber slides on the communication itself with the second FSO head. The Optical Signal is converted to an electric power station using an AT-MC 102 XL AC power converter. The converter is also signed with a standard ethernet cable that is connected to a network card located in the LOS lab. The optical fiber type requires an active electro-optical converter and also initialization of the FSO line to obtain the received optical signal. In this case, a converter is used, which is terminated by the RS232 port.

Due to different carrier frequencies, the FSO and RF links have different channels. The short wavelengths used in the FSO junctions are seriously affected by dispersion due to fog and atmospheric scintillation, while RFs are particularly sensitive to rain scattering. The use of both systems to overcome these problems can lead to increased reliability and speed [8]. When configuring FSO link switching, the transmitter and receiver together select either the FSO or RF channel for data transmission. The FSO high speed connection is only selected if the channel conditions allow reliable communication, otherwise all data is sent over the RF channel. The transmitter and receiver must be coordinated by feedback on selecting the right channel for transmission.

The idea to overcome the disadvantageous switching is to coordinate the transmission of data in both connections by channel coding. The data is encoded with one LDPC code with a portion of the codeword divided into FSO and RF connections, and the speed is set according to the instantaneous channel conditions. Although this technique improves due to hard switching, channel conditions must be known on the transmitter and receiver and complex fine decoding is required, which is difficult at FSO data rates. Therefore, the use of Raptor coding for FSO/RF hybrid links is considered [8].

III. ROLE OF THE MINI PC IN HYBRID FSO/RF SYSTEM

The main task of the mini computer is to monitor weather conditions along the atmospheric channel using sensors such as temperature, humidity sensor, wind speed sensor and others. Measurement results are used as inputs for machine learning methods, for processing and prediction of received optical performance, which is the implementation of RSSI parameter in hybrid FSO/RF line switching (Fig. 3).

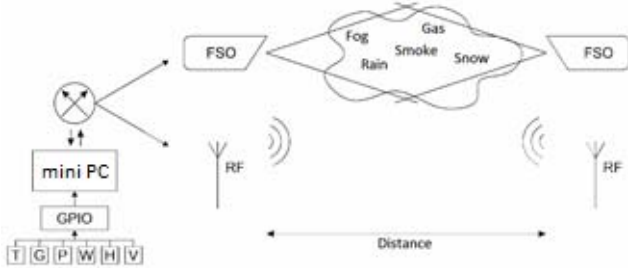


Fig. 3 Scheme of the Hybrid FSO/RF System

The main software components of the software system to enhance the availability and reliability of FSO systems include, in addition to the control scripts, MySQL databases, as well as the Nginx web server, which provides the ability to interact with instantaneous measured values of individual physical quantities (Fig. 4).

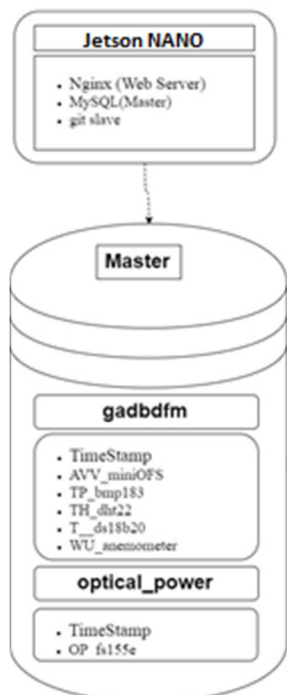


Fig. 4 Organization of database system

The main advantages of the selected mini-computer NVIDIA Jetson Nano include computation performance, open operating system, GPIO bus (the possibility of connecting analog and digital sensors), all of these features are advantageous for implementation in an experimental system to analyze the availability and reliability of FSO systems.

TABLE I. TECHNICAL SPECIFICATION OF THE MINI PC

Parameter	Value
GPU	128-core Maxwell
CPU	Quad-core ARM A57 1,47 GHz
Memory	4 GB 64-bit LPDDR4 25,6 GB/s
Video Encode	9x 720p @ 30 (H.264/H.265)
Video Decode	18x 720p @ 30 (H.264/H.265)
Connectivity	Gigabit Ethernet, M2 Key E
USB	4x USB 3.0, USB2.0 Micro-B
Peripherals	GPIO, I ² C, I ² S, SPI, UART
Mechanical	69 mm x 45 mm

Manufacturer Support JetPack Software Development Kit (SDK) provides a complete Linux environment for Jetson Nano based on Ubuntu with accelerated graphics, support for CUDA Toolkit and libraries like cuDNN and TensorRT. The SDK also includes the ability to native install popular open source Machine Learning (ML) frameworks such as PyTorch, TensorFlow, Keras, Caffe, and MXNet, along with frameworks for the development of computer vision and robotics, such as OpenCV and Robot Operating System (ROS) [9].

By applying machine learning to a mini computer, it is possible to define an automated process that extracts patterns from data. We use supervised computer education to create models used in predictive data analysis applications. Machine Supervision techniques will automatically learn the model of the relationship between a set of descriptive functions and a target function based on a set of historical examples or instances. This model can then be used to predict new cases [10].

Machine learning algorithms work by searching for a set of possible prediction models for the model that best captures the relationship between descriptive characters and the target function in the dataset. The obvious criteria for managing this search are finding models that match the data. There are at least two reasons why finding consistent models is not enough to teach useful prediction models [11]. First, for large data sets, it is likely that the data will have noise and prediction models that are consistent with the noisy data will make incorrect predictions. Second, in the vast majority of machine learning applications, the training set represents only a small sample of a possible set of instances in the domain [12]. As a result, machine learning is a poorly posed problem. A badly posed problem is a problem for which it is not possible to identify a unique solution using only available information.

IV. APPLICATION OF MACHINE LEARNING TO MINI PC

Machine learning techniques are widely used today for many different tasks. Different data types require different methods. The task of machine learning in a hybrid FSO/RF system is to predict the strength of the received optical signal, which is influenced by weather conditions. A form of machine learning to the gradient boosting on the decision tree, which works by gradually training more complex models, is designed to maximize the accuracy of predictions. Gradient enhancement is particularly useful for predictive models that analyze organized data and categorical data.

Enhancing decision tree is one of the most efficient ways to make models. The combination of gradient enhancement with decision trees enables results in applied applications with structured data. There are two types of errors that can lead to inappropriate inductive bias: overfitting and underfitting [13]. Overfitting occurs when the prediction model selected by the

algorithm is too simplified to represent the underlying relationship in the dataset between descriptors and the target function. Underfitting, on the other hand, occurs when the prediction model chosen by the algorithm is so complex that the model is too closely associated with the data file and becomes sensitive to noise in the data. It is an effort to use techniques to get your own model in an iterative way. During the first iteration, the algorithm learns the first tree to reduce errors. If the model has a significant error, it is not a good idea to create large decision trees, including preliminary data. The second iteration (Fig. 5), in which the algorithm learns one tree to reduce the first tree error. The algorithm repeats this procedure until the appropriate model.

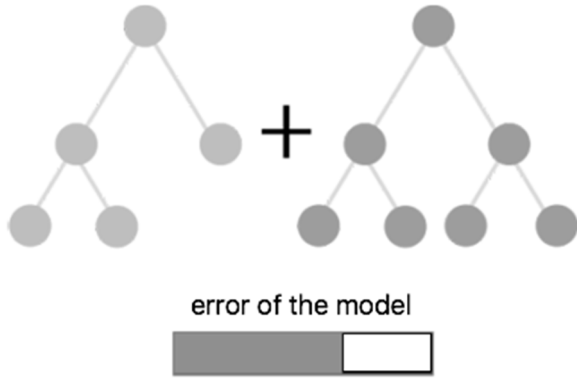


Fig. 5 Second trees include error

Gradient Boosting (GB) is a method of implementation in the absence of an objective function. The normal pre-classification approach utilizes logarithmic loss, while regression optimizes with mean squared error (MSE). Each GB step combines the following steps:

- Gradients of the loss function are calculated for each input object.
- Prediction of loss gradients using decision tree [14].

Handling inputs in requires decision trees in an iterative way. For the sake of simplification, we can consider classification trees that can be easily described. Finding the next decision tree requires calculating all possible element distributions so that the element value is less than any predefined value of all elements in the data. Then select the one that improves the loss function by the greatest value. After creating the first distribution, the next distribution in the decision tree is selected as follows: the first distribution is fixed and the next distribution is selected according to the first. These steps are performed until the entire decision tree is created [15].

Training set is used to create a description that can be used to predict previously unseen examples. Attributes (sometimes called fields, variables, or functions) are typically one of two types: nominal (values are members of an unordered set) or numeric (values are real numbers). Attribute a_i , it is useful to mark its domain values as:

$$\text{dom}(a_i) = \{v_{i,1}, v_{i,2}, \dots, v_{i,|\text{dom}(a_i)|}\} \quad (1)$$

Instantial space (a set of all possible examples) is defined as the Cartesian product of all input attribute domains:

$$X = \text{dom}(a_1) \times \text{dom}(a_2) \times \dots \times \text{dom}(a_n) \quad (2)$$

Universal instance space U is defined as the Cartesian product of all inbound attribute domains and target attribute domains:

$$U = X \times \text{dom}(y) \quad (3)$$

A training set is an instance consisting of a set of m -tics. Formally, the training file is marked as $S(B) = (\langle x_1, y_1 \rangle, \dots, \langle x_m, y_m \rangle)$ where $x_q \in X$ and $y_q \in \text{dom}(y)$ [16]. It is usually assumed that the training sets are generated randomly and independently according to some fixed and unknown common probability distributions of D above U . Generally it is a generalization of a deterministic case when the supervisor classifies the set using the function $y = f(x)$ [11].

To create a prediction w using a node in a tree that contains a set of instances, the weight of the instance in the current tree is predicted:

$$SSE(y, \hat{y})^t = \frac{1}{2} \sum_{i=1}^I (y_i - (\hat{y}_i + w))^2 \quad (4)$$

Overwritten in terms of residues:

$$SSE(y, \hat{y})^t = \frac{1}{2} \sum_{i=1}^I (r_i^2 - 2r_i w + w^2) \quad (5)$$

After simplifying the substitution for $\sum_{i=1}^I r$ as R

$$SSE(y, \hat{y})^t = \frac{1}{2} \sum_{i=1}^I r_i^2 - R w + \frac{1}{2} n w^2 \quad (6)$$

The above equation gives the training loss of a set of instances in a leaf. The optimal leaf weight w^* is given by setting:

$$\frac{dSSE(y, \hat{y})^t}{dw} = 0 \quad (7)$$

By substituting for the loss function for the simultaneous iteration and simplifying, we get the effect of predicting w^* in the given leaf

$$SSE(y, \hat{y})^t = \frac{1}{2} \sum_{i=1}^I r_i^2 - \frac{1}{2} \frac{R^2}{n} \quad (8)$$

Different termination conditions and methods for selecting the next node lead to different learning approaches. The best-known of these approaches are leaf and depth schemes. Using the leaf node approach, the algorithm divides the area to achieve the best improvement in the lossy function, and the process continues until a fixed number of leaves is obtained [18]. The depth approach algorithm builds tree-by-level until a fixed-depth tree is created. Choosing such decision trees has several advantages over conventional trees:

- Simple installation.
- Effectively implement on CPU.
- The ability to produce very fast model applicators.
- Provide benefits for many tasks due to regularization.

Shared memory can be used for aggregation, which is characterized by its speed. Since one limitation is the size of shared memory, it is convenient to group several functions into one to achieve maximum computational performance. The learning algorithm of the classical decision tree is computationally demanding. In order to find further distribution, we have to evaluate the number of observation times for different distribution conditions. This leads to a large number of possible differences for large data sets through continuous inputs and, in many cases, overfitting. Fortunately, it is possible to significantly reduce the number of divisions

that we need to consider. We can make a rough approximation of the input functions. The easiest way to quantify is to use the quantiles of the input element distribution.

The algorithm solves feature problems and also supports category-based functions. Handling categorical elements is one of the challenges of machine learning. The most commonly used technique for solving categorical predictors is single hot-coding. The original function is removed and a new binary variable is added for each category. This approach also has disadvantages:

- Deep decision trees need to be built to restore dependencies in the event of high cardinality. This can be solved by a hash trick, but such an approach significantly affects the resulting quality.
- Such an approach does not work for values of unknown categories such as values that do not exist in the learning data set.

An interesting comparison is the computational speed of a given mini-computer compared to other commercially available products. The test was performed using the TensorFlow library. The results were compared on the following platforms: Coral Dev Board, NVIDIA Jetson Nano, Raspberry Pi.

TABLE II. BENCHMARK OF MINI-PC

Dev Board	Time (ms)
Coral Dev Board	273,2
Nvidia Jetson Nano	302.8
Raspberry Pi	520,6

The basic assumption is that samples with examples of low gradient training have minor training errors and are well trained. In order to maintain the same data distribution, a constant multiplier for data cases with small transitions is introduced in the calculation of the gain of information. This achieves a good balance between reducing the number of data cases and maintaining accuracy for learned decision trees and reducing the time needed to make decisions.

V. CONCLUSION

The main disadvantage of external (external) FSO systems is the relationship of signal and transparency of the atmospheric transmission channel. Atmospheric transparency is primarily based on weather phenomena, which are mainly in the form of rain, snow and fog. The availability and reliability of any technology is an essential prerequisite for its use in real life. This problem is compensated by the high-frequency line technology, which is most fog-resistant, and therefore represents the biggest problem for FSO systems. Due to the high impact of weather on FSO transmission, a back-up RF line is needed, which is less prone, but much slower. For this reason, a hybrid FSO/RF line is required which can be switched. To create such a system, it is necessary to analyze the conditions by collecting weather data to obtain characteristics of the variables that affect the transmission channel. A suitable hardware system coupled with machine learning methods makes communication more efficient.

ACKNOWLEDGMENT

This work was supported by Cultural and Educational Grant Agency (KEGA) of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the project No. 023TUKE-4/2017 and by the Slovak Research and Development Agency under the contract no. "APVV-17-0208 - Resilient mobile networks for content delivery".

REFERENCES

- [1] V. Brazda, O. Fiser, and J. Svoboda, "FSO and radio link attenuation: meteorological models verified by experiment," in *Free-Space and Atmospheric Laser Communications XI*, A. K. Majumdar and C. C. Davis, Eds. SPIE-Intl Soc Optical Eng, sep 2011.
- [2] P. Singal, S. Rai, R. Punia, and D. Kashyap, "Comparison of different transmitters using 1550nm and 1000nm in FSO communication systems," *International Journal of Computer Science and Information Technology*, vol. 7, no. 3, pp. 107–113, jun 2015
- [3] S. A. Al-Gailani, A. B. Mohammad, R. Q. Shaddad, and M. Y. Jamaludin, "Single and multiple transceiver simulation modules for free-space optical channel in tropical Malaysian weather," in 2013 *IEEE Business Engineering and Industrial Applications Colloquium (BELAC)*, April 2013, pp. 613–616.
- [4] J. Tóth, "High availability and reliability in wireless optics using data analytics techniques" - 2017. In: *SCYR 2017*. - Košice : TU, 2017 S. 118-119. - ISBN 978-80-553-3162-1
- [5] M. L. B. Riediger, R. Schober, and L. Lampe, "Blind detection of on-off keying for free-space optical communications," in *2008 Canadian Conference on Electrical and Computer Engineering*, May 2008, pp. 001 361–001 364
- [6] F. Nadeem, V. Kvicera, M. Awan, E. Leitgeb, S. Muhammad, and G. Kandas, "Weather effects on hybrid FSO/RF communication link," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 9, pp. 1687–1697, dec 2009 [Online]. Available: <https://doi.org/10.1109/JSA.2009.091218>
- [7] M. I. Petkovic, g. T. Dordevic, and D. N. Milic, "Ber performance of IM/DD fso system with oob using apd receiver," *Radioengineering*, vol. 23, no. 1, apr 2014.
- [8] R. Gupta and P. Singh, "Hybrid fso - rf system: A solution to atmospheric turbulences in long haul communication," *International Journal of Scientific Engineering Research*, vol. 5, no. 11, nov 2014.
- [9] D. Franklin, "Jetson Nano Brings AI Computing to Everyone," *NVIDIA Developer Blog*, 23-Apr-2019. [Online]. Available: <https://devblogs.nvidia.com/jetson-nano-ai-computing/>. [Accessed: 15-Sep-2019].
- [10] M. Joly, S. Sarkar, and D. Mehta, "Machine Learning Enabled Adaptive Optimization of a Transonic Compressor Rotor With Pre-Compression," *Volume 2C: Turbomachinery*, Nov. 2018.
- [11] T. Hastie and J. M. Chambers. "Statistical models." *Statistical Models in S*. Routledge, 2017. 13-44.
- [12] H. Gavin. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.
- [13] M. Brzoska, "Modellierung zwischen „overfitting“ und „underfitting“," *Rationale Entscheidungen unter Unsicherheit*, pp. 152–156, 2018.
- [14] G. Mittag and S. Möller, "Detecting Packet-Loss Concealment Using Formant Features and Decision Tree Learning," *Interspeech 2018*, Feb. 2018.
- [15] X. Zhong and W. Li, "A Decision-Tree-Based Multi-objective Estimation of Distribution Algorithm," *2007 International Conference on Computational Intelligence and Security (CIS 2007)*, 2007.
- [16] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees Classifiers—A Survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, 2005.
- [17] O. Maimon and L. Rokach, "Introduction to Supervised Methods," *Data Mining and Knowledge Discovery Handbook*, pp. 149–164.
- [18] A. S.bhatlavande and A. A. Phatak, "Data Aggregation Techniques in Wireless Sensor Networks: Literature Survey," *International Journal of Computer Applications*, vol. 115, no. 10, pp. 21–24,

Training Neural Networks with Computer Generated Images

János Hollósi
Department of Information Technology
Széchenyi István University
Győr, Hungary
hollosi.janos@sze.hu

Áron Ballagi
Department of Automation
Széchenyi István Egyetem
Győr, Hungary
ballagi@sze.hu

Abstract—At the Széchenyi István University we develop an autonomous racing car for the Shell Eco-marathon. One of the main tasks is to create a neural network which is segment the road surface, the protective barriers and other components of the race track. The difficulty of this task, that there is no a right dataset for this special issue. Only a limited size dataset available, therefore, we would like to expands this dataset with computer generated training images, which comes from a virtual city environment. In this work we want to examine the effect of computer generated images on the efficiency of different neural networks. In the training process real images and computer generated virtual images are mixed in several different ways. After that, three different neural network architecture for road surface and road barrier detection are trained. Experiences shows how to mixing datasets and how they can improve efficiency.

Keywords—neural network, computer generated dataset, virtual dataset, autonomous vehicle.

I. INTRODUCTION

The Shell Eco-Marathon is an international competition held by the Shell corporation. This event is a challenge for university students to design, develop, build and drive the most energy-efficient race cars. Our University's race team, the Szenergy team is a successful participant of the Shell Eco-marathon for over 10 years. Two years ago the Shell introduced the Autonomous Urban Concept (AUC) Competition, which is an additional competition for self-driving vehicles at the Shell Eco-marathon. In the Autonomous Urban Concept Competition, participants have to complete five different challenges, like parking on a dedicated parking rectangle, obstacle avoidance on a straight track, drive up one lap on the track in autonomously and so on.

The long term goal is to be ready for the AUC competition. This paper focus on road surface and road barrier detection. We would like to do the detection by neural network based approach, because neural networks are one of the best tools for visual based detection problems, like image segmentation. Nowadays, many high-performance neural network architectures are available, such as AlexNet by Krizhevsky et al. [1], VGG Net by Simonyan and Zisserman [2], GoogleNet by Szegedy et al. [3], Fully Convolutional Network by Shelhamer et al. [4], U-Net by Ronneberger et al. [5], ResNet by He et al. [6] and Pyramid Scene Parsing Network by Zhao et al. [7]. Training neural networks requires large amount of data. However, in this case, we do not have a sufficient number of training samples. For example we do not have any training image about the road barriers and it takes a lot of time and energy to generate and annotate real world data. Our idea is to use computer simulation environment to generate training data for this task. There have been some attempts that use

virtual generated data to train neural networks. Peng et al. [8] demonstrated CAD model based convolutional neural network training for object detection. Tian et al. [9] presented a pipeline to build virtual scenes and a virtual datasets for neural networks. They proved that mixing virtual and real data to train neural network for object detection helps to improve the performance. Židek et al. [10] showed a new approach to object detection using neural networks trained by virtual model based datasets. Our paper would like to show the effect of computer-generated training data on the learning process for different network architectures.

The paper is structured as follows: in section II we describe the simulation environment what we use for generating training data; in section III we show our neural network architectures; in section IV we describe the training process of the networks; in section V we show results and experiences; finally, section VI presents our conclusion.

II. SIMULATION ENVIRONMENT

For the training data generation we created a computer simulation environment. Unreal Engine is used to create small virtual cities with many traffic situations. The Unreal Engine is a game development engine, developed by the Epic Games. It is mainly used to develop computer games, but nowadays it can be used in more and more fields, like autonomous vehicle development. Where we can use it to create training data, or create a test environment for simulating self-driving vehicles and so on. This is a very good and energy-efficient method, when we would like to use a special dataset, which does not exist or is not accessible.

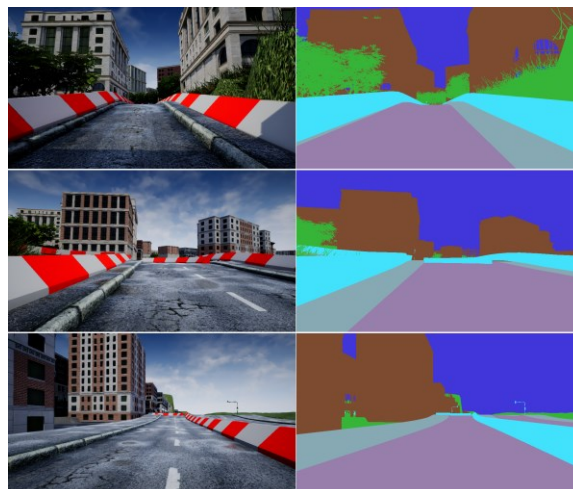


Fig. 1. Sample images from our virtual image dataset.

In the simulation, we needed to make training images for the neural networks, that look like they were taken from a vehicle, and for each image, it was necessary to create a segmented version of it. Fig. 1 shows some sample from the virtual dataset. In the segmented image each object has a different and unique color. The virtual city environment is created in two ways: First, each virtual object has the real, correct texture, and second, every object has a solid color for texture. We developed a simple script for the virtual environment, which takes one picture with the real texture, and after that, takes an another picture with the solid color texture, at a given point. The vehicle was manually driven through the streets of the virtual cities, while our script captured image pairs.

In the virtual world the following object categories are defined:

- building,
- road,
- sidewalk,
- barrier,
- vegetation,
- sky.

In this research we would like to focus only on the road surface and the road barriers, because this objects are the most important for the Shell Eco-marathon challenge.

III. APPLIED NETWORK ARCHITECTURES

Three different neural network architectures are implemented in this work: FCN, U-Net and PSP Net. All neural networks are designed to image segmentation, where the size of input image is $256 \times 512 \times 3$, and the size of the output is $256 \times 512 \times 1$. Every networks are trained to road surface and road protection barrier segmentation.

A. FCN

The FCN [4] architecture is based on a fully convolutional layers, where the basic idea is to extend effective classification networks to segmentation tasks.

Our Fully Convolutional Network architecture are shown in fig. 2. Let γ a block, which contains a convolution, batch normalization and ReLu activation layers. Let b_1 includes two γ blocks and one max pooling layer, and let b_2 is similar to b_1 , but it contains three of block γ . Based on all these let B_1 contains two b_1 and one b_2 , $B_2 = b_2$, and B_3 is based on one b_2 and three γ blocks, where the last three γ block is the fully connected layer. The x block contains a convolution and a batch normalization, the y contains only a ReLu activation, and the z contains a ReLu and a Softmax activations. In this implementation all convolution layers are 3×3 size, except the three fully connected layer in the B_3 block. This convolutional layers have 7×7 size. In the B_1 block the first two convolution contains 64 filters, the third and fourth layers contains 128 filters, and the last three convolution block includes 256 filters. All convolution layer in the B_2 block contains 512 filters. The first three convolution in the B_3 contains 512 filters, and the fully connected layers based in 4096 filter.

B. U-Net

The U-Net [5] neural network architecture was originally created for biomedical image segmentation. It is based on Fully Convolutional Network, where the network can be divide into two main parts: the downsampling and the upsampling block.

Our implementation are shown in Fig. 3. Let γ block is the same as in FCN network. The D_1 block contains two γ block and one max pooling. The D_2 is the same as D_1 . The D_3 , D_4 and B blocks are the same. They contain three of the γ block and one max pooling layer. The U_1 is the first upsampling block. It contains a transposed convolution, batch normalization, ReLu activation, and two of the γ block. The U_2 , U_3 and U_4 blocks contains the same elements as U_1 , but the U_4 block includes one extra convolution and a softmax activation. In the U-Net network all convolutions are 3×3 sized, and transposed convolutions are 2×2 sized. The number of convolutional filters are the following. All convolutional layers in D_1 have 64 filters, in D_2 have 128, in D_3 have 256 and in D_4 and B have 512. The upsampling part is very similar. In U_1 have 512, in U_2 have 256, in U_3 have 128 and in U_4 have 64 filters.

C. PSP Net

The Pyramid Scene Parsing Network [7] is the best architecture on the ImageNet [11] Scene Parsing Challenge in

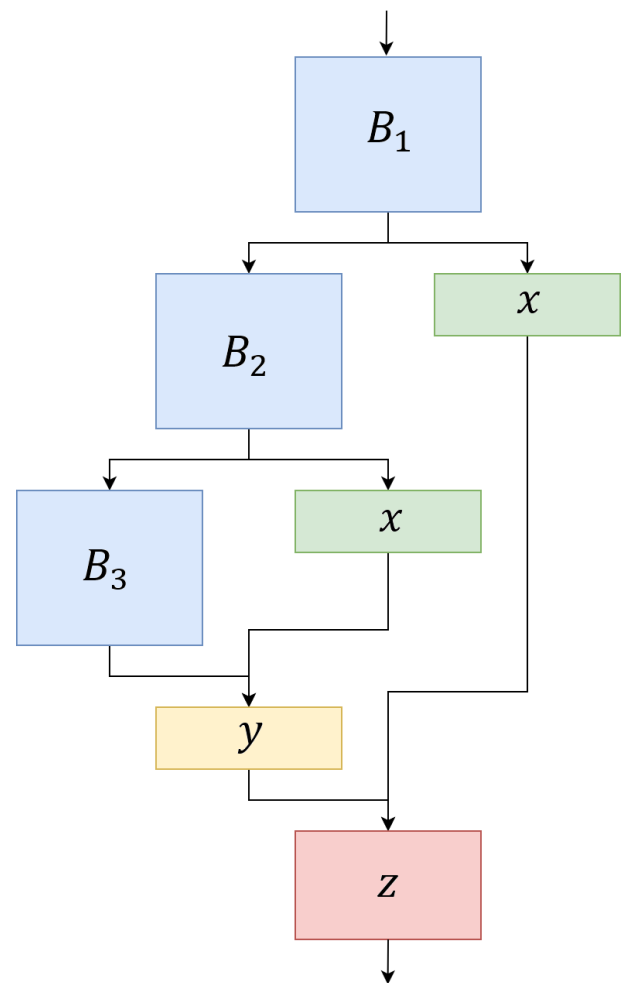


Fig. 2. FCN architecture.

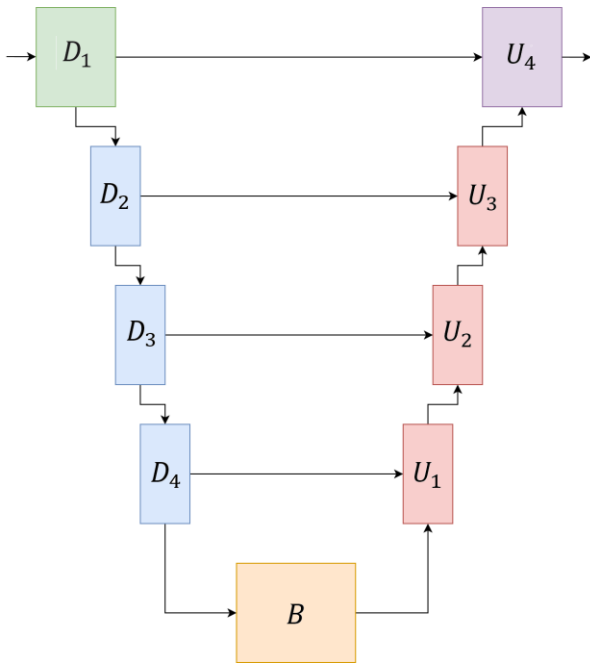


Fig. 3. U-Net architecture.

2016. The main building block of the PSP network is a pyramid pooling module, where the network fuses features under four different pyramid scales.

Our Pyramid Scene Parsing Network based architecture are shown in Fig. 4. In the PSP Net we use the same c block, which is defined in III. A. The first block is a B_1 , that is structured as follow: three of γ block and a max pooling. This is followed by a series of the two main building block: C is the convolutional block and I is the identity block. Both blocks consist of two parallel layer. In the C block the first layer contains two γ block, a convolution and a batch normalization. The second layer includes a convolution and a batch normalization. This two layer are merged at the end of the block, and after that we make a ReLU activation. The I block is very similar to the convolutional C block, but, in this case the second layer is empty. The end of the first layer is merged with the originally input of the block. After the B_1 block the Pyramid Scene Parsing Network architecture contains four convolutional block, and after every convolutional block, it includes some identity block, in the following quantities: after the first convolutional, two, after the second convolutional, three, after the third convolutional, five, and after the last convolutional block, it contains two identity block. Thereafter comes the pyramid pooling block. Our pyramid block is build up by four main layers, the P_1 , P_2 , P_3 and P_4 . Every layer contains a series of average pooling and convolution layers. P_1 contains one, P_2 contains two, P_3 contains three, and P_4 contains four series of this. After that, the result of the pyramid pooling layers are concatenated. In the B_2 block we can found one γ , a dropout, a convolution a transposed convolution and finally a softmax activation. In the B_1 block all convolution has 3×3 size. In the C and I blocks every first and third convolution has 3×3 size, and every second convolution has 1×1 size. In the B_2 the size of the first convolution is 3×3 , and the second is 1×1 . The transposed convolution's size is 16×16 . The first two convolution in the B_1 has 64 filters, and the last one has 128. In the first C block and the first two I blocks contains 64, 64,

256 filters, the second C block and the next three I block has 128, 128, 512 filters, the third C block and the five I blocks after that has 256, 256, 1024, and the fourth C block with the last two I blocks has 512, 512, 2048 filters.

IV. TRAINING THE NEURAL NETWORKS

We would like to improve the accuracy of neural networks with computer generated training images, which comes from the virtual city environment. We made some mixed datasets, which contains real world images and computer generated virtual images. The real world images is come from the Cityscapes dataset [12]. The Cityscapes dataset is a large-scale dataset for semantic segmentation. The dataset contains 5000 fine annotated images, which is created in 50 different cities with various weather conditions. The number of object classes is 30, like road, sidewalk, person, vehicles, traffic light, terrain, sky and so on, but in this research we focus only to the road surface detection. In this research only the Cityscapes dataset is used, because this is one of the biggest dataset for traffic situations. The computer generated images is come from the simulation environment, described in II.

We created five different datasets from Cityscapes and virtual image collection. Table 1. shows how we mixed this two collection. Our goal is to use a minimal size of data from a real dataset, and when we change the amount of virtual images, observe how it affect the efficiency of the neural networks.

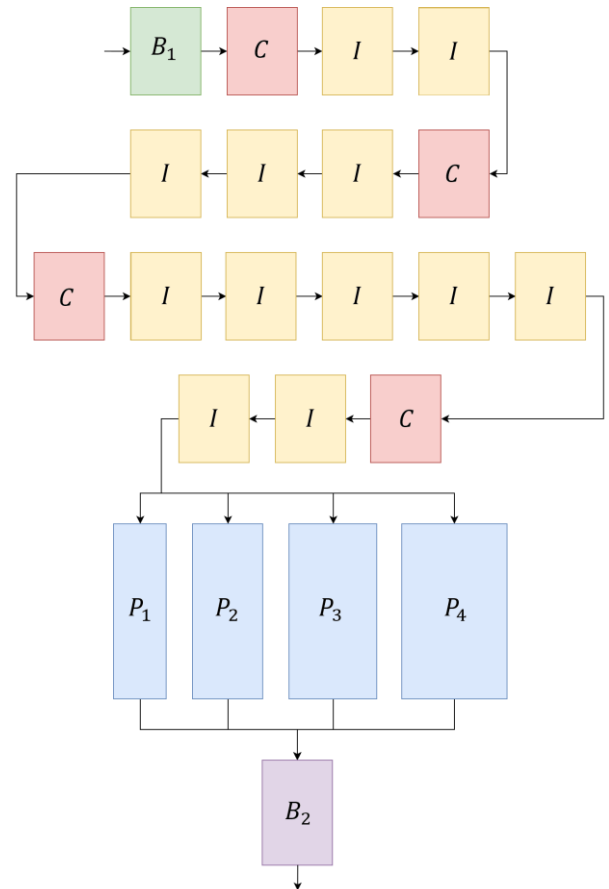


Fig. 4. PSP Net architecture.

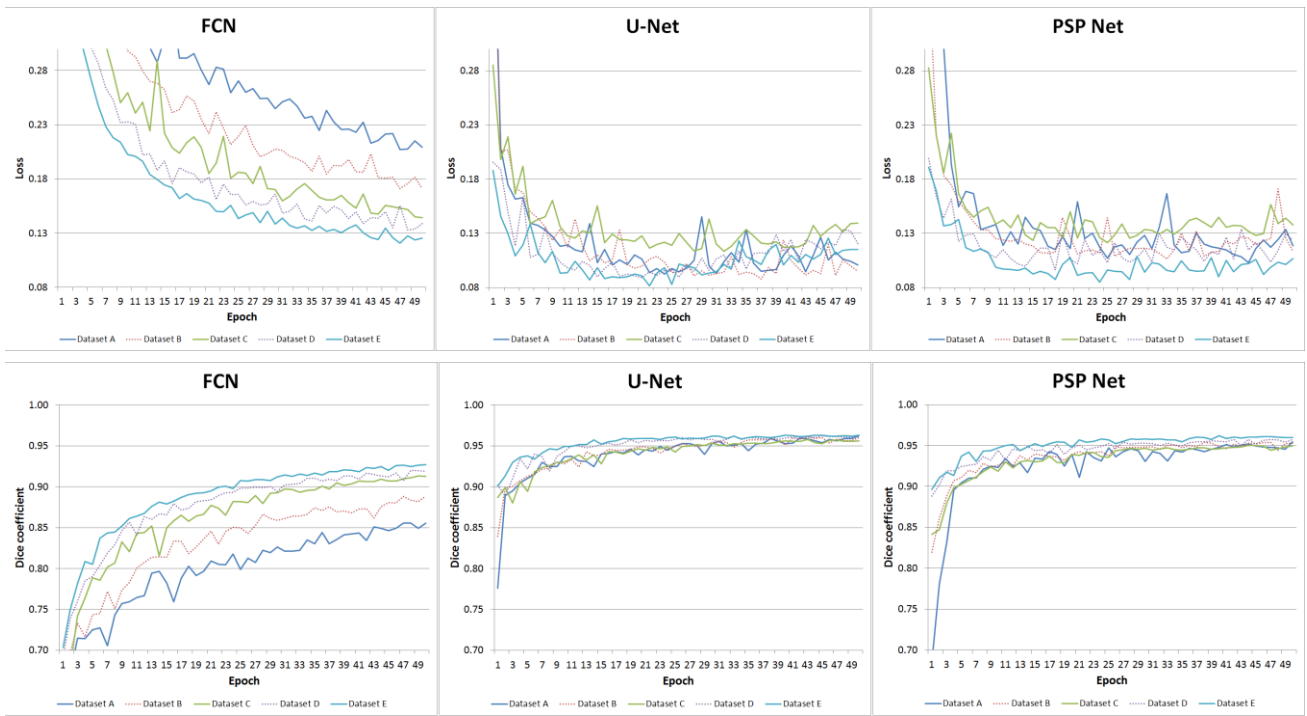


Fig. 5. Performance of the three neural networks on the five different datasets under the road detection training process.

TABLE I. NUMBER OF VIRTUAL AND REAL WORLD IMAGES IN OUR ROAD DETECTION DATASETS

Dataset name	Train set		Validation set	
	Virtual	Real	Virtual	Real
Dataset A	0	500	0	125
Dataset B	500	500	0	250
Dataset C	1500	500	0	500
Dataset D	1500	1000	0	625
Dataset E	1500	1500	0	750

Dataset A contains only real world images, therefore this is the basic dataset, the other datasets have been compared to this. Dataset B already contains the same virtual images as real images. Here we want to observe the introduction of virtual images how change the initial efficiency. Dataset C contains three times more virtual images than dataset B. If the number of virtual images is much more than the number of real-world images, the efficiency may be reduced. We would like to investigate this. In dataset D and dataset E the number of real images was increased. Here we investigated the effect of increasing the number of real-world images on efficiency. Adam optimizer is used in training with 10^{-4} learning rate and 5×10^{-4} learning decay. For the objective function categorical crossentropy is used and dice coefficient is measured.

V. RESULTS & EXPERIENCES

We tried to examine the road surface detection efficiency, while the composition of the dataset is modified. For examining the changes of efficiency, the most useful datasets are the dataset A, dataset C, and dataset E. The dataset A is the basic dataset, it contains only a small set of real world images.

Compared to the dataset A, the dataset C is based on dataset A, but it contains three times more virtual images as real images. This dataset shows how performance changes, when we mixing virtual world images to a small sized dataset. In the dataset E the size of the collection is expanded. This shows how much better the efficiency of a larger mixed dataset. Fig. 5 shows the validation accuracy and loss over the training process of the road detection, and Fig. 6 shows the best loss and dice coefficient values for the road surface segmentation. FCN networks architecture is much simpler than U-Net and PSP Net architectures. Hence the efficiency of the FCN network on the dataset A is a little worse, than the other networks. U-Net and PSP Net architectures are very robust

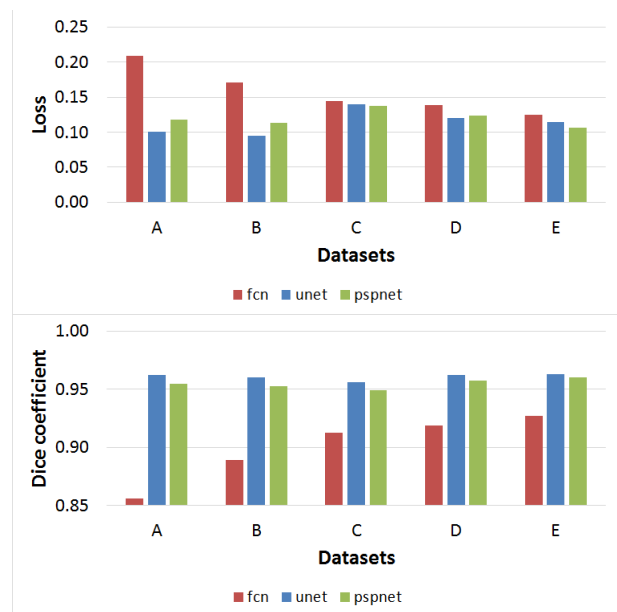


Fig. 6. Comparison of the final loss and accuracy values in road detection process.

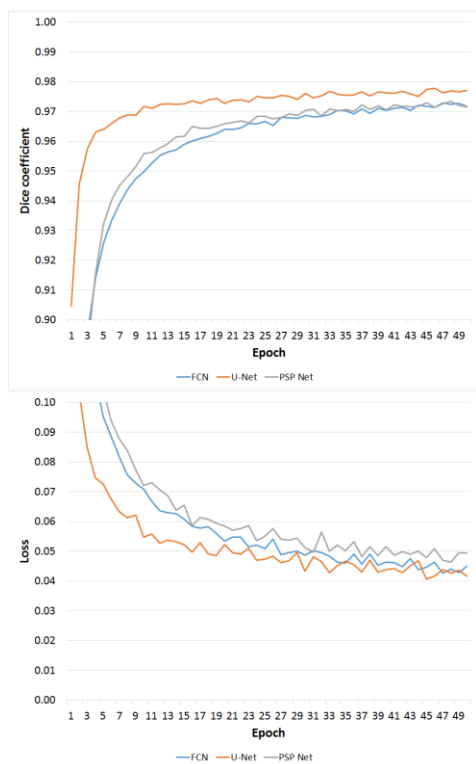


Fig. 7. Barrier detection performance.

and complex, therefore mixed datasets do not increase significantly the efficiency on this architectures. However, for simpler networks like FCN, this method improves the efficiency. For the road barrier detection we created a dataset, which contains 1572 virtual training images and 131 real images for validation. Fig. 7 shows the performances of the barrier detection. Only virtual images were used for train the barrier detector networks. This would not have been possible in case of road surface detection, because the road surface is too complex. The texture of the road barrier is very simple, therefore it is possible to recognize it based on only virtual images.

We would like to use environment detection system aboard a low-budget racing car, where the hardware resources available are limited and the detection must work in real time with high detection accuracy. Therefore the neural network should be designed to be as simple as possible. If the network architecture is too simple, it is more difficult to train for complex recognition tasks. Moreover we don't have large size dataset about racetrack, road barriers and so on. In this case, it is helpful to be able to train simpler networks, like FCN, with virtual datasets for higher efficiency. Experience has shown that using three times more virtual images improves the efficiency of road surface detection, and for road barrier detection it enough to use only virtual images.

VI. CONCLUSION

This paper presents how to use computer-generated virtual images to train artificial neural networks, when the size of

available real world images is limited. We investigated three different neural network architectures, as FCN, U-Net and PSP Net, and trained this networks with mixed datasets. We showed that virtual images improve the efficiency of neural networks. Our experience shows when the texture of the objects is simple, like the road barrier, it is sufficient to use only virtual image based training datasets. This work may help for us to creating an efficient environment detector for the Shell Eco-marathon, where we must to detect special objects without real-world datasets.

ACKNOWLEDGEMENT

The research presented in this paper was carried out as part of the "Talent Management in Autonomous Vehicle Control Technologies (EFOP-3.6.3-VEKOP-16-2017-00001)" project in the framework of the New Széchenyi Plan. The completion of this project is funded by the European Union and co-financed by the European Social Fund.

REFERENCES

- [1] G. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," 25th International Conference on Neural Information Processing Systems (NIPS), Nevada, USA, vol. 1, pp. 1097-1105, 2012.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd International Conference on Learning Representations, San Diego, USA, 2015.
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet and S. reed, "Going deeper with convolutions," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015.
- [4] E. Shelhamer, J. Long and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, 2017.
- [5] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," In: N. Navab, J. Hornegger, W. Wells, A. Frangi (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol. 9351, pp. 234-241, Springer, Cham, 2015.
- [6] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 770-778, 2016.
- [7] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, 2017.
- [8] X. Peng, B. Sun, K. Ali and K. Saenko, "Learning deep object detectors from 3D models," IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 1278-1286, 2015.
- [9] Y. Tian, X. Li, K. Wang and F. Wang, "Training and Testing Object Detectors with Virtual Images," IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 2, pp. 539-546, 2018.
- [10] K. Židek, P. Lazorík, J. Pitel and A. Hošovský, "An Automated Training of Deep Learning Networks by 3D Virtual Models for Object Recognition," Symmetry, vol. 11, pp. 496-511, 2019.
- [11] O. Russakovsky, J. Deng, H. Su et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, pp. 211-252, 2015.
- [12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, 2016.

Enhancing Team Interaction and Cross-platform Access in Web-based Collaborative Virtual Environments

Marián HUDÁK, Štefan KOREČKO, Branislav SOBOTA

Department of Computers and Informatics

Faculty of Electrical Engineering and Informatics

Technical University of Košice, Slovak Republic

Marian.Hudak.2@tuke.sk, Stefan.Korecko@tuke.sk, Branislav.Sobota@tuke.sk

Abstract—Utilization of web-based Collaborative Virtual Environments (CVE) enhances user's interaction over large geographical distances. Deploying a variety of VR technologies increases demands to develop VR applications which would run on various systems and devices. Otherwise, user's access is limited by the supported operating system. Nowadays, concerning the quality and safety of web technologies, it is possible to provide Global Collaborative Virtual Environments (G-CVE) with cross-platform support. This paper introduces the LIRKIS G-CVE web-based system, built on top of the Networked-Aframe framework. The architecture of LIRKIS G-CVE is presented, together with its innovative Smart-client interface (SCI) component, which utilizes smart devices, such as smartphones and tablets, as input devices. The advantage of SCI over the standard input component of the Networked-Aframe is demonstrated via a series of experiments.

Keywords—*virtual reality, web-based, virtual environment, virtual collaboration, human-machine interaction,*

I. INTRODUCTION

Virtual Reality (VR) provides an effective approach to the development of simulations and various activities in Virtual Environments (VE) [1]. Nowadays, VR applications can offer an immersive VE that may be constructed faster than a physical environment [2]. During the last years, there were increasing requirements to produce a shared VE for Virtual Collaboration that may be available for multiple users in real-time. Therefore, Collaborative Virtual Environments (CVE) were proposed to involve groups of participants in solving variety of tasks and also with the ability to operate over a network [3]. However, most of them are intended to provide a local-based connection without worldwide access, which limits their purpose.

Deployment of Web-based CVEs can provide global collaboration without limitation in geographical location of its users [4]. Web-based CVEs possess several strengths increasing its usability and demand for Virtual Reality. Concerning cross-platform access, current VR technologies are facing hardware diversity with demands to establish software compatibility to run on various operating systems. A web cross-platform can simplify user access and reduces development time [5].

In contrast with technological progress, optimization of 3D graphics and inputs for cross-platform VR remains

insufficient. However, these difficulties can be resolved by integration of dynamic device recognition and entity component interfaces utilization, with the capability of device performance and peripherals identification. This paper presents LIRKIS Global - Collaborative Virtual Environment (LIRKIS G-CVE) [6], while taking enhancement of its integration with various platforms and multi-user interaction in real-time into account.

The rest of the paper starts with a brief comparison of LIRKIS G-CVE with other related solutions (section II). Sections III and IV deal with LIRKIS G-CVE: Section III describes its architecture and section IV focuses on a newly developed user interface component, called SCI, which uses smart devices as input peripherals. Section V experimentally evaluates SCI by comparing it to the previous solution, provided by an underlying framework. The conclusion summarizes the current enhancement and equipment of the LIRKIS G-GCVE and presents ideas of its future use.

II. RELATED WORK

Web technologies simplify the creation of CVEs through various graphics libraries and standards supporting JavaScript, HTML and CSS. The work most related to the LIRKIS G-CVE is [7], where architecture for web-based collaborative 3D virtual spaces using DOM synchronization is presented. Both [7] and our work employ the same VR web framework, A-Frame¹ for building immersive CVEs. The work [7] implements the architecture for sharing 3D virtual spaces through a server aimed at the DOM and object synchronization while using a NoSQL database. In the LIRKIS G-CVE, the object synchronization is provided over client-server data channels via WebSockets. This process is more effective in asynchronous real-time connection when the number of clients and objects varies during collaboration. In the case of user interaction, our solution employs smart-devices, such as multipurpose VR controllers, while the work [7] deploys the A-Frame standard inputs and focuses on visual output for head-mounted displays.

Another related architecture, called SimCEC [8], provides multi-user CVE through Unity3D application, executed on various web browsers. CVE distribution is ensured by a dedicated cloud server. Similar to our work, the system evaluates the user's activity during the collaboration and

¹ <https://aframe.io/>

processes results in real-time. The SimCEC concentrates upon Medical CVE, however our solution is able to cover an extensive range of CVEs and collaborative applications. The LIRKIS G-CVE supports server-side scripting through online IDE without installation of external tools, such as Unity3D. In order to enhance user interaction, the SimCEC includes a hand tracking to correct positioning of user's hands. Our system involves a cross-platform VR interface, which is able to process smartphone sensor data to control 3D interaction.

The VirtualOulu system [9] uses a 3D virtual collaborative model of town in web-based immersive VR. The system is intended for laptop computer utilization instead of wearable devices. On the other hand, LIRKIS G-CVE is designed to be adaptable for a variety of displays and VR equipment. Both solutions depend on Three.js library to render CVEs in real-time.

III. LIRKIS G-CVE

The LIRKIS G-CVE [6] represents an immersive web-based VR system to provide multi-user connection. The system's architecture utilizes Entity-Component-System (ECS), which offers high flexibility to create various VR applications and extensions of CVE in ranging demands on complexity (Fig.1). Each entity can contain multiple, fully reusable components, which can be mixed according to the final use. The communication between the system and users is based on client-server architecture to share data over the network. The LIRKIS G-CVE provides client-side rendering (CSR) because of processing more complex virtual environments with a variety of visual effects, lights and shades.

A. Remote Web Server

The main part of the LIRKIS G-CVE is a remote web server, which is responsible for providing communication among clients and also mediating networked entities in shared CVEs. The server uses three JavaScript frameworks namely Node.js, Express.js and Networked-Aframe (NAF), which support full backend services. The Node.js provides real-time parallel client connection and manages the asynchronous client-server data stream. Usage of Express.js was considered as important to handle HTTP requests from clients during the connection. The NAF mediates all of the Networked-Aframe components (NAC) and synchronizes them among all clients. The NAC consist of a shared scene, its entities, and features as Transferring entity ownership (TEO) and Networked entity manipulation (NEM). The TEO creates a relation, which allows the user to take control of the object, while the NEM supports variety of object translations.

The communication between client and server is based on the WebSocket protocol. Each CVE is accessible under specific URL address for clients. The server is processing all of the clients' data, including their IDs and Avatar coordination. After a client is connected, the server requires its Client data and then broadcasts it to all of the active connections. The amount of transferred data between clients depends on complexity of CVE according to provided interaction, number of shared entities and their details. Client-server communication is bandwidth sensitive, which prevents server overloading by inactive networked entities. This feature was utilized to eliminate data transmission and reduces the

same data replication concurrently. When a client leaves a CVE, its connection is terminated by the server.

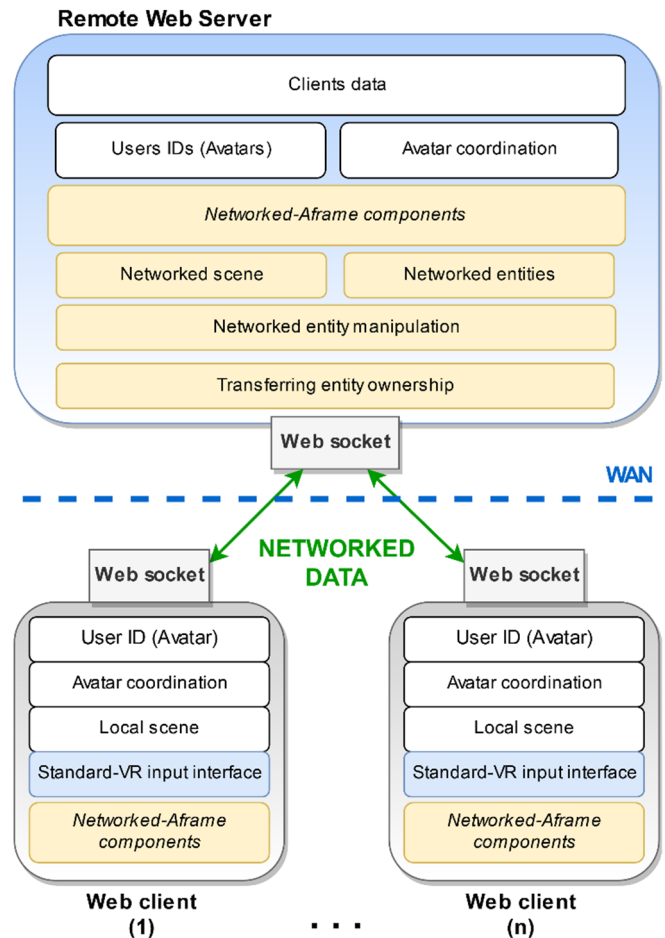


Fig. 1. The LIRKIS G-CVE system architecture.

B. Web-client Interface

Our intention was to provide a web-client interface with asynchronous CVE rendering. In the implementation stage, the A-Frame technology was chosen with respect to its guaranteed cross-platform support and scalability through ECS. The A-Frame [10] works under the Three.js 3D JavaScript Library to render 3D content on the web. By using the Asynchronous *JavaScript* (AJAX), it is possible to render all of the changes simultaneously and without reloading the page.

The Web-client interface consists of several components responsible for its integrity and interaction. The most important are User ID and Avatar Coordination, which are sent as first after the client connects to the server. The Local scene component contains a virtual environment with all of 3D entities and surrounding objects composed through NAC. To support user interaction, each Web-client utilizes an integrated Standard-VR input interface that collects all of the components responsible for user movement and object manipulation. These components are able to process data from different VR controllers and standard inputs. However, each type of input is conducted by its own component. Therefore, if CVE requires user movement by touch and VR Gamepad, it is necessary to use two separate components. Each kind of

interaction increases the number of input components, which is likely to cause a problem when users don't provide them.

IV. MULTIPURPOSE VR INTERFACE

As mentioned in section III, the LIRKIS G-CVE utilizes ECS to provide reusable functionality mixed with different entities. However, the *Standard-VR input interface* contains a large number of components serving each type of input. Therefore, manipulation with a single entity by different inputs increases the number of active components concurrently. This causes their faulty integration when a user desires to utilize multiple input VR devices at the same time. It also reduces the client's performance. This may negatively affect the user interaction. An extension of multipurpose interface component can improve the same functionality as the previous ones but simplify implementation of user interaction.

A. Multipurpose Smart-client interface

Considering this issue, the multipurpose VR interface component called *Smart-client interface (SCI)* (Fig.2) was proposed. SCI would eliminate the number of VR input devices by using smart devices, namely smartphones and tablets. We decided to use smart devices because of the range of peripherals they provide: a variety of sensors, a touch screen, and potential haptic feedback. Smartphones also offer more functionality than dedicated VR controllers.

Generally, each *Web-client (user)* can access the LIRKIS G-CVE through different platforms and devices such as Desktop Computers, VR Headsets, as well as with Microsoft HoloLens, seamlessly. These devices are able to provide visual feedback and some of them support standard inputs.

Extending the LIRKIS G-CVE of the SCI can positively affect the Web-client's performance in the rendering of visual output. The main role of SCI is to handle the computing of user interaction. In the same way, it is responsible for 3D object manipulation similar to VR hands, tracked controllers and 3D pointers. These features allow the user to interact with 3D objects more naturally.

B. Features Implementation

In the first stage of implementation, we decided to extend an already existing NAF component, which was responsible for mapping device sensors through JavaScript events. The main intention was to gather all of the data that the smart-device can provide. Therefore, the Dynamic Device Recognition (DDR) was considered as helpful to recognize all of sensors and features available in the smart device. To provide 3D object manipulation, it was important to acquire data from the IMU of the smart device.

The accelerometer was also utilized to trigger sensor data stream when a user shakes the device. This feature was proposed to conduct device activity, while the user is wearing a VR headset and can't see his hands. The gyroscope and magnetometer were important to ensure 3D object orientation. To enable touch inputs for object selection and grabbing, the graphical user interface with variety of sliders and buttons was considered as necessary.

The second stage of implementation was focused on communication between a Web-client and a SCI over the network (Fig.2). On the server side, we decided to implement the pairing functions that attach the SCI to its Web-client by their IDs. The SCI shares its position, rotation and geometry with the server, which provides these data to other networked entities. In the same meaning, the Web-client is only observing all of SCI activities. Considering that SCI uses a smart device, it is able to handle all interactions autonomously and it does not overload the Web-client performance.

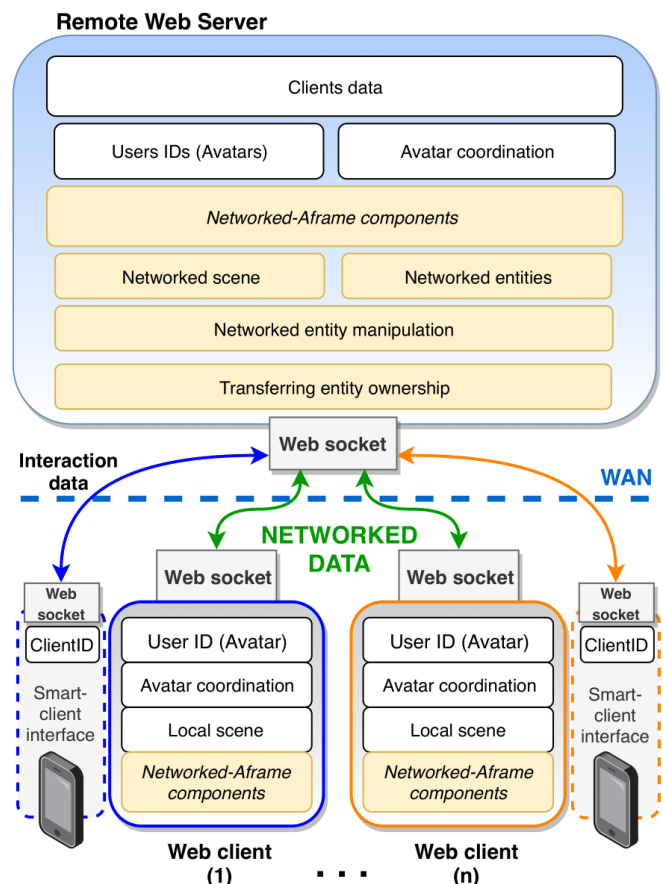


Fig. 2. The LIRKIS G-CVE extension of the Smart-client interface (SCI).

The third stage focused on manipulation with networked entities and interaction through SCI. The first type of interaction was implemented using ray-casting method (Fig.3 a). The Raycaster [11] includes a 2D line extended from the user towards the direction where it checks its intersection with surrounding objects (Fig.3. b). Each of them can be selected and then manipulated. The second type of interaction involves 3D object collision detection between the user and the surrounding scene (Fig.3 c). Each collision is processed by 3D volumes called bounding boxes (3D colliders) that are wrapped around objects [12]. The 3D collider can be formed into various shapes such as a primitive box, a cube, a sphere or a cylinder. Utilization of 3D colliders positively affects object selection and manipulation through 3D pointers and VR Hands, which can collide with surrounding objects.

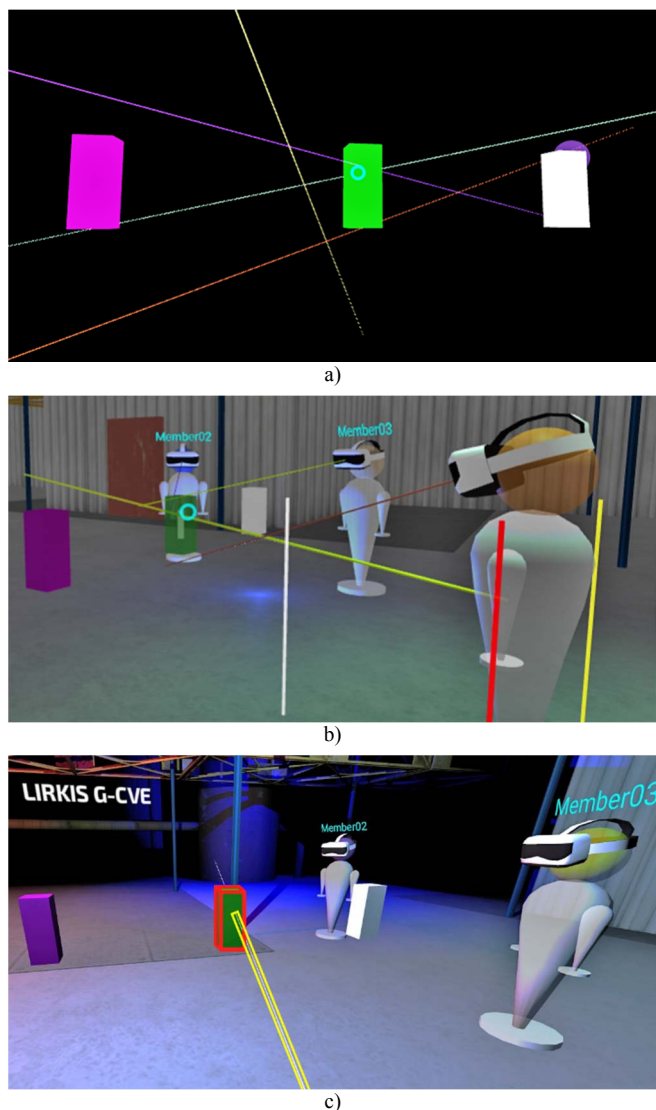


Fig. 3. The LIRKIS G-CVE smart-client interface : SCI raycasters intersection (a), hand and gaze-based interaction (b) and bounding volumes in collision detection (c).

V. PERFORMANCE EVALUATION ON DIFFERENT PLATFORMS

After finalizing the implementation, we performed several experiments to evaluate results of LIRKIS G-CVE with the recently deployed interface. The purpose of the experiments was to compare previous Standard-VR input interface and the newly created Smart-client interface (SCI). We decided to prepare tests of both solutions with three end-user devices: an ASUS FX504 SERIES notebook, a VR Headset with a SAMSUNG J5 2017 SM-J530F smartphone and MS HoloLens (1st generation). The notebook has been equipped with Intel Core i5-8300H CPU (2.30GHZ), 8.00GB RAM, NVIDIA GeForce GTX 1050 graphics card with 4GB video RAM (GDDR5), 5400RPM HDD and used 64bit Windows 10 Home operating system.

In the first experiment, we measured the framerate during user interaction. The second experiment evaluated the response time between interface input action and visual output rendering. Each experiment consisted of 1000 trials and used the same CVE with 300 000 polygons.

A. FPS measuring experiment evaluation

The results of the first experiment, shown in Fig.4, describe the framerate in Frames Per Second (FPS) for both interfaces. Each of them was tested with corresponding VR controller. A VR gamepad was used as the Standard-VR input interface, while the SCI was used with a smartphone. When the Standard-VR input interface was employed, measured FPS values were lower compared to employing the SCI. The most significant difference of FPS rates was observed while using MS HoloLens rendering device, where the average difference was 21 FPS. The least significant difference occurred with the usage of the notebook. This can be explained by its adaptable computing performance. In this case, the difference was 13 FPS. In all cases, SCI performed better than the Standard-VR input interface.

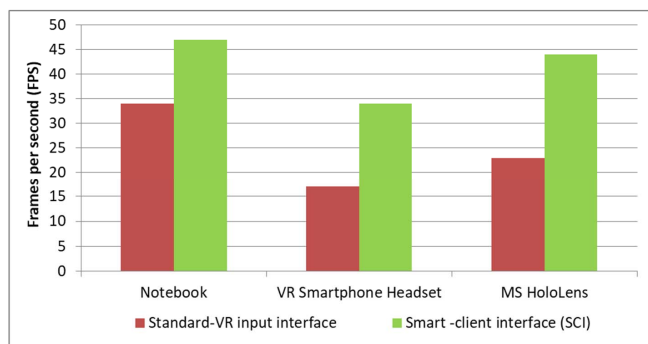


Fig. 4. Comparison of rendering performance.

B. Measuring rendering response time

The second experiment compared the response time between interface input action and Web-client's visual output reaction. The same VR devices and interfaces were utilized as in the first experiment. The averages of response times for all 1000 trials are shown in Fig.5. For each trial, the measurement was carried out in the following way: First, the interface input action was performed and the time t_i of the action was recorded. Second, the time t_e of an event, triggered by the input action on the rendering device was recorded. Then, the response time has been computed as $t_e - t_i$. The measurements confirm that the Web-client rendering response was lower when using SCI compared with the Standard-VR. The most significant difference was observed when the VR Smartphone headset was utilized; with the average time of 72 ms. Similarly to the results of the first experiment, the least significant difference occurred when the notebook was used with a time of 21 ms.

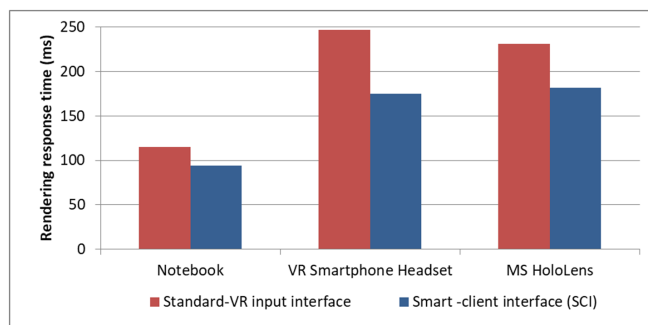


Fig. 5. Comparison of measured response time.

Based on the evaluation, the SCI interface yields positive results of testing with various rendering devices. The testing of SCI proved increased rendering performance on each of the rendering devices. Considering the fact, that the interface utilizes a smart device, it can be deployed without the need for special calibration and setup. The currently implemented multipurpose interface improves natural interaction and simplifies user access to the LIRKIS G-CVE.

CONCLUSION

In this paper, we have presented the extension of the LIRKIS G-CVE system with improving VR inputs and user interaction. We assume that virtual collaboration can be more beneficial in the future due to its simplicity and accessibility. Despite the available technologies, it is necessary to consider cross-platform support as an important factor in providing CVE worldwide. Utilization of smart devices makes the virtual reality much more accessible to ordinary people. This was the main reason for our decision to implement the smart mobile interface as a VR controller for CVE interaction. Based on the experiments results, we may consider the development and deployment of SCI into the LIRKIS G-CVE a success and significant improvement. In the future, we would like to concentrate research on the multiuser object manipulation between numerous users in real-time. This option can bring many benefits in controlling 3D objects, shared between users, which should enhance virtual collaboration over long distances. SCI is currently used in several VE, developed for LIRKIS G-CVE, including a virtual hand application to be used for motor function rehabilitation.

ACKNOWLEDGMENT

This work has been supported by the APVV grant no. APVV-16-0202 "Enhancing cognition and motor rehabilitation using mixed reality".

REFERENCES

- [1] S., Sharma, P., Devreaux, D., Scribner, J., Grynovicki, & P., Grazaitis. Megacity: A Collaborative Virtual Reality Environment for Emergency Response, Training, and Decision Making. *Electronic Imaging*, 2017, pp.70-77.
- [2] A., Correia, B., Fonseca, H., Paredes, P., Martins & L., Morgado. Computer-simulated 3D virtual environments in collaborative learning and training: meta-review, refinement, and roadmap. In *Handbook on 3D3C Platforms* (pp. 403-440), 2016, Springer, Cham.
- [3] E., Poppe, R., Brown, J., Recker & D., Johnson. Improving remote collaborative process modelling using embodiment in 3D virtual environments. In *Proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling-Volume 143*, 2013, pp. 51-60.
- [4] D., Seo, B., Yoo & H., Ko. Webizing collaborative interaction space for cross reality with various human interface devices. In: *Proceedings of the 23rd International ACM Conference on 3D Web Technology*, ACM, 2018, p. 12.
- [5] W. S., El-Kassas, B. A., Abdullah, A. H., Yousef & A. M., Wahba. Taxonomy of cross-platform mobile applications development approaches. In: *Ain Shams Engineering Journal*, 8(2), 2017, pp.163-190.
- [6] M., Hudák, M., Sivý. Web-based collaborative virtual environments to support cross-platform access. In: *Poster 2019 International student scientific conference*, Prague, 2019, pp.178-182.
- [7] C., Gadea, D., Hong, D., Ionescu & B., Ionescu. An architecture for web-based collaborative 3D virtual spaces using DOM synchronization. In *2016 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2016, pp. 1-6., IEEE.
- [8] P. V., Paiva, L. S., Machado, A. M. G., Valença, T. V., Batista & R. M. Moraes. SimCEC: A collaborative VR-based simulator for surgical teamwork education. *Computers in Entertainment (CIE)*, 2018, 16(2), 3.
- [9] T., Alatalo, T., Koskela, M., Pouke, P., Alavesä & T., Ojala. VirtualOulu: collaborative, immersive and extensible 3D city model on the web. In: *Proceedings of the 21st International Conference on Web3D Technology*. ACM, 2016. p. 95-103.
- [10] Aframe: A web framework for building virtual reality experiences homepage (2019), <https://aframe.io/>
- [11] T., Nukarinen, J., Kangas, J., Rantala, O., Koskinen & R., Raisamo. Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, 2018, p. 86. ACM.
- [12] J., Allard, F., Faure, H., Courtecuisse, F., Falipou, C., Duriez & P. G., Kry. Volume contact constraints at arbitrary resolution. In: *ACM Transactions on Graphics (TOG)*, 2010, 29(4), pp. 82.

Realtime attack environment for DDoS experimentation

Ladislav Huraj
Department of Applied Informatics
University of SS. Cyril and Methodius
 Trnava, Slovakia
 ladislav.huraj@ucm.sk

Marek Šimon
Department of Applied Informatics
University of SS. Cyril and Methodius
 Trnava, Slovakia
 marek.simon@ucm.sk

Abstract— **Distributed Denial of Service (DDoS) attack has been one of the most serious threats to network security for years. An attacker floods a target server using a large number of compromised computers in order to radically consume victim's resources (CPU, memory). Although the impact of DDoS attacks on services has been done in several simulations and emulations, the real attack environment reflects more accurately the results of validation and testing of the services. Moreover, the real environment of DDoS attacks can bring other new issues that cannot be seen in the simulation. The article describes design, implementation and testing of proposed DDoS attack testbed. The effort was to create a testbed for the purpose of testing a pseudo-botnet consisting of thousands of nodes in order to try to move as close as possible towards the power of real DDoS attacks. The experimental results prove the potential of proposed real-time DDoS attack testbed and show the impact of targeted attack on a particular victim machine and network.**

Keywords—*DDoS attacks, network security, real-time environment*

I. INTRODUCTION

A distributed denial of service (DDoS) attack is an attack carried out by multiple compromised/misused sources distributed across the Internet to flooding victim server or network in a way that victim server cannot handle it. The attacker using the compromised devices sends fake or malicious traffic to the victim servers to overwhelm them with traffic and rendering them incapable of serving legitimate users.

DDoS attacks have been on the rise in recent years and show no sign of slowing up. The 14th Annual Worldwide Infrastructure Security Report [1] reveals that in 2018 a DDoS attack topped 1 Tbps in size for the first time ever. Moreover, with a global max attack size increase of 273 percent in 2018, there are clearly significant implications and new challenges for enterprises and service providers alike. This disappointing statistic indicates that volumetric DDoS attacks, especially reflection/amplification, are a continuing challenge.

The article describes the design, implementation and testing of proposed DDoS attack testbed and tackles the problem of shared cyber-physical infrastructures under attack. The performance of a DDoS attack is examined in real time under different scenarios. The DDoS testbed was scaled up to hundreds and thousands of attacking nodes of different types to provide better experimentation to get as close to 'real world' scale as possible with as much scientific integrity as possible. On the other hand, easy handling by system administrators was desired. An orchestration as the automated configuration,

coordination, and management of computer systems and software was used to manage tasks and data in the distributed heterogeneous environments. In our case, Ansible orchestration tool has been employed. Ansible is a simple, reliable, powerful open source orchestration tool, working in collaboration with a strong community of partners and offers both configuration management and also high-end orchestration [2].

Moreover, Ansible structure operates on client-server topology, which naturally follows the structure of the botnet network used in DDoS attacks. Our previous experiments confirm that it is possible to create testing environment of pseudo-botnets by using special software for ad hoc grids establishing [3,4]. However, the ad hoc grid software has to be installed on individual nodes; it is not always possible if the nodes are for example routers or IoT devices. Even building such ad hoc grid on smart phones is problematic [5]. On the other hand, using the Ansible tool makes it easy to build a test environment on heterogeneous devices on the network.

The designed attack environment can serve as a tool for network administrators and researchers to test and examine the condition of security of networks and to analyze compensatory measures to DDoS attacks in real conditions in real-time which also proves the experimental results of the case study in the paper.

The article is organized as follows: Section II stages related work, Section III describes design of the DDoS testbed, Section IV provides experimental setup and results of performed attacks, and finally, section V reveals conclusion and future work of the research.

II. RELATED WORK

Several DDoS testbeds have been designed by many scientists. Testbeds can contain virtual and/or physical components, and may be general-purpose or highly specialized for a specific focus area. Four basic strategies are used to validate DDoS testbed [6-8]:

1. *Mathematical models*: symbolic modeling and subsequent mathematical validation of given platforms, systems, and applications.

2. *Simulation*: a simulation as an imitation of the behavior of a specified system over time using simulation software, e.g. Omnet++, ns-2, ns-3, Opnet, GloMoSim or Qualnet3. As the process of simulation is relatively easy and cheaper than the complete real implementation, analyzing of variety of experiments can be done at low cost. For example, network simulation analyzes the behavior of a network by computing the association between different network devices such as hubs, switches, routers, access points, links etc. On the

This work was supported by the grant VEGA 1/0145/18 *Optimization of network security by computational intelligence* and the grant KEGA 011UMB-4/2017 *Increasing competencies in work with high performance computing ecosystem*.

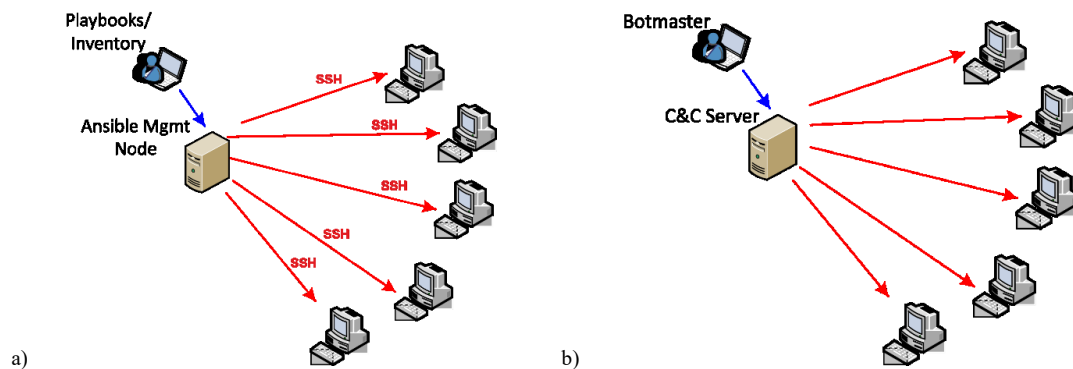


Fig. 1 Similarity of architecture of a) Ansible and b) Centralized C&C botnet

contrary, simulators cannot perfectly model all the details of the simulated platform [9].

3. *Emulation*: is a combination of simulation and real systems. Some components are simulated (e.g. traffic, links) and some are applied on real hardware (e.g. notes). Much greater fidelity is capable in emulation than simulation because the functionality of one system in terms of another system is duplicated.

A significant difference is in the time of an experiment; while simulation proceeds in virtual, simulated time, the emulation proceeds in real time. An example of systems for emulation are environments like: DETER, ns-3, WIL (WAN-in-Lab), Emulab [6,9].

4. *Real Systems*: Although emulators as well as simulators are beneficial tools for the performance evaluation of tested platforms, unrealistic assumptions and oversimplified models can lead to inaccurate results. Real systems give realistic conditions, real platforms and applications and are proven to be best for network based experimentations [6]; e.g. PlanetLab or GENI.

Especially for DDoS attacks, testbeds based on live testing under DDoS attack compare to either simulation- or emulation-based tools can produce more realistic results [9]. Still most of the existing testbeds only relate to testing of DDoS attacks with low false alarm rate, high detection rate or with secure and isolated environment and are of academic interest only [10].

In the paper, rather than on an isolated environment, the proposed testbed focuses on controlled attack and on real-world scheme under continuous supervising.

III. DESIGN OF DDOS TESTBED

The structure of the proposed testbed was inspired by a structure of centralized C&C botnet, Fig. 1b). Centralized C&C botnet relies upon a single centralized C&C resource to manage compromised machines (bots) where attacking botmaster delivers commands to the machines via C&C (command and control) channel and coordinates attacks and fraudulent activities. In this kind of botnet the central C&C point issues instructions directly to each bot [11].

Ullah et al. in [12] categorize the botnet architecture into four categories where the centralized C&C botnet is the first one. The categories depend on the way how a botnet controls network and how it hides its detection. Besides the centralized C&C botnet, there are categorizes P2P botnet, hybrid botnet,

and botnet combining the http protocol with P2P. All these kinds of botnet are often used for a DDoS attack. For the proposed testbed design, the similarity with the first category is essential where the resemblance with structure of Ansible tool was utilized and the orchestration feature from Ansible was applied to organize the DDoS attack.

Ansible remotely manages fleet of machines from a central point called Controller Machine where Ansible software is installed, Fig. 1a). The remote machines are connected over the SSH protocol but no additional software should be installed or run on them since Ansible utilizes an agentless structural engineering without installing an agent managed machines but instead it pushes, executes and removes modules on the machines using only Python and a few other packages. To control the machines, Ansible retrieves the information about managed machines from the inventory file and then executes the code over SSH from playbook file consisting of tasks represented by a YAML file [13,14].

Comparison of structure of centralized C&C botnet and Ansible orchestration tool is illustrated on Fig. 1. The above diagram reveals that the architecture of Ansible tool is comparable to the centralized C&C botnet design. The centralized C&C botnet in Fig.1b) involves botmaster, C&C server and a group of bots. A group of bots receives and responds to commands from the C&C server that acts as a rendezvous mechanism for orders from the botmaster. By contrast, the Ansible tool Fig.1a) contains playbooks/inventory, Ansible management node, and group of remoted machines. The playbook is used to manage configuration which can be deployed on the remote machines and the inventory contains all the machines that are managed in the groups. The playbook is designed in simple human-readable format. An Ansible management node is a machine where the Ansible software is installed and which is responsible for configuration of all remoted machines [2].

In centralized C&C botnet, botmaster delivers commands to the compromised machines through C&C channel, and controls them. The C&C infrastructure executes commands from botmaster to bots and allows the botmaster to guide and control whole botnet. C&C infrastructure interconnects the botnet components for data transferring among them. For the botnet is crucial to have this relation stable in order to work efficiently [15]. In Ansible environment, the Ansible management node will be connected during the execution of a human-readable Ansible playbook via SSH on each host

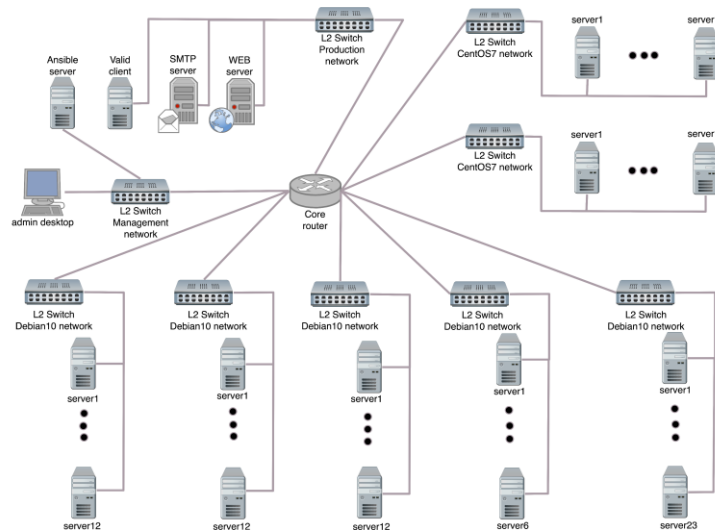


Fig. 2 Structure of the testbed based on Ansible environment

defined in the inventory, and remotely execute each task defined in the playbook [16].

However, when comparing the behavior of the environments, the main difference is that while main aim of the botnet is attacking the victims, the main aim of the Ansible tool is to assist to automate the job processing. Additionally, encrypted SSH communication in Ansible is for confidentiality protection; while in the botnet the encrypted C&C communication is used to make difficult to detect any clear behavior of the botnet at the network level from payload analysis [17].

The process of creating the testbed itself consisted of the following 8 steps: (i) KVM hypervisor preparing for the powerful servers, (ii) creating virtual machine with Ansible environment (Python and a few other packages, CentOS7), (iii) creating virtual machines (32 VMs with CentOS7, 36 VMs with Debian10, Debian10 installation on a physical server and its cloning by Clonezilla SE and PXE on 28 additional servers), (iv) creating network infrastructure for the experiments, (v) migration of a copy of the production web server (CentOS7 LAMP) to the hypervisor into test environment. Anonymization of sensitive data on the production web server was used, (vi) migration of a copy of the production mail server (Debian10, postfix, dovecot) to the hypervisor into test environment. Anonymization of sensitive data on the production mail server was used, (vii) creating scripts (shell, python, Perl) generating and measuring the attacks, (viii) creating Ansible playbooks to launch attacks and to measure them. The creation process resulted in establishing of testbed infrastructure illustrated on Figure 2 that provides almost ten thousand different IP addresses usable for carrying out the attacks.

It should be noted that since Ansible tool is not directly designed for DDoS attacks it was necessary to address some minor issues with this environment to achieve the desired goal during Ansible preparation. For example, one of the Ansible strategy as a way to control play execution is strategy *free* which allows each node to run until the end of the play simultaneously as fast as it can without waiting for all nodes. However, it is important to watch out the race condition

problem, i.e. not to allow a new task to start until the previous block of tasks is completed.

On the other hand, the proposed testbed environment based on Ansible tool fulfils all the requirements for a DDoS testbed described in [18]: (i) an experiment should be specified, saved and replayed; (ii) an experiment should be deployed, run and stopped; (iii) the performed DDoS attack should be monitored during the run as well as the logs should be archived for later repetition or examination.

IV. EXPERIMENTAL SETUP AND RESULTS

The testbed environment consisted of 97 servers, Fig. 2. The number of physical servers was 29 running OS Debian10, Intel(R) Core(TM) 2 Quad CPU Q8400 @ 2.66GHz, 2 GB RAM. In addition, servers based on Kernel-based Virtual Machine (KVM) module that allows the kernel to function as a hypervisor were used, explicitly 32 servers running CentOS7, one core Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 1GB RAM and 36 servers running Debian10, Intel(R) Xeon(R) CPU E3-1230 v3 @ 3.30GHz, 1 GB RAM. Each server emulates 100 virtual nodes to increase the number of unique IPs in the network and so the final number of all unique IPs reached the value of 9 700. The bandwidth of all links in the network topology was 1 Gbps.

The copies of two real production servers were used for test purposes, a web server for the first two scenarios and a mail server for the last attack. It means that there are no default settings set but the operation is adapted to real run, e.g. mail server uses real spam filters and antivirus software and controls each incoming email or defense mechanisms are installed. Moreover, effects of the attacks were discussed with servers' administrators and have been proven as useful to create more appropriate countermeasures against server attacks.

The first scenario was a SYN flood attack targeting the server TCP/IP stack. Attack is focused towards a listening port of the server where an enormous amount of TCP-SYN requests with spoofed IP addresses of sources is sent by the attacker. Consequently, the computational resources of the victim server are exhausted because of the number of open connections and server services become unavailable. A

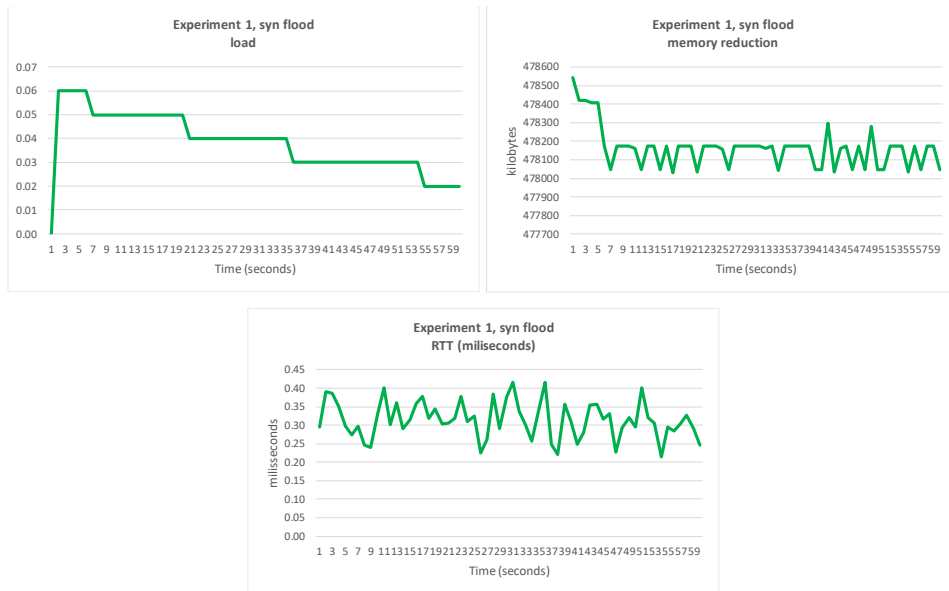


Fig. 3 Metrics values for SYN Flood attack

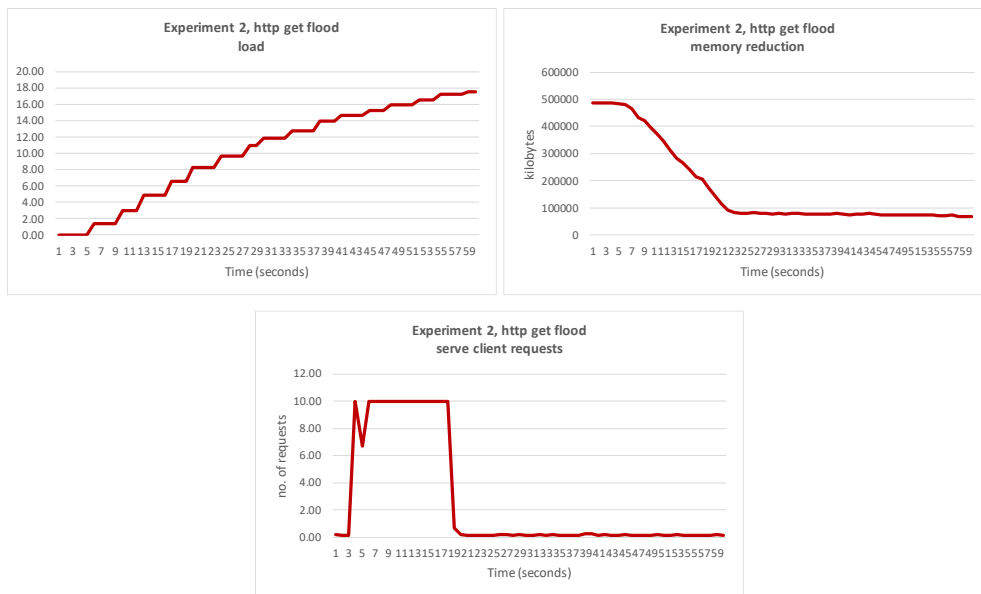


Fig. 4 Metrics values for Http Get Flood attack

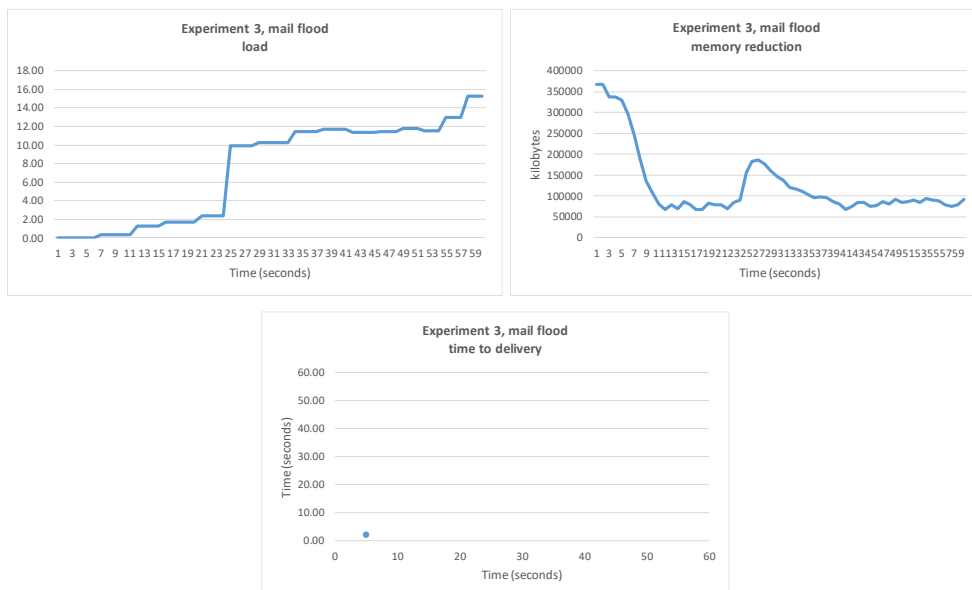


Fig. 5 Metrics values for Mail Flood attack

Scapy was used for the attack. Scapy is able to build or decode packets, send and capture them, as well as to match requests and replies. It combines the functions of scanning, probing, tracerouting, unit testing, attacking and network discovering [19]. OS CentOS7, one core Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 2GB RAM was running on the victim web server. Fig. 3 shows the result of the SYN flood attack. The duration of SYN flood attack was 60 seconds.

The HTTP Get flood was led from proposed environment to target server as the second experiment for 60 seconds. Overwhelming of the server's computing resources is done through basic GET request conducted in application layer from attacker to victim server. A load testing and benchmarking tool for HTTP server ApacheBench was used for the attack in the performed experiment [20]. The same victim web server as in previous experiment was employed. Fig. 4 displays the result of the HTTP Get flood attack.

The last type of performed attacks was SMTP mail flooding, which tried to overwhelm the mail server, i.e. to send an enormous number of successive emails to mail server in an attempt to collapse the target mailbox or affect a non-tolerable delay of the mail server. Emails are sent with a different message to go through the spam filter. The attack can also result in loss of connectivity of whole network [21]. For the SMTP mail flood, the Scapy tool was used for 60 seconds. The victim mail server configuration was the same as for web server, i.e. OS CentOS7, one core Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz, 2GB RAM. Fig. 5 shows the result of the SMTP mail flood attack.

As main metrics in the analysis, the CPU load and memory usage of the victim server, and average serve rate for all performed experiments were taken into account. *CPU load* is a measure of the amount of computational work, i.e. of tasks running, of tasks waiting for CPU and of tasks blocked. *Memory usage* of the victim server is the amount of main memory utilized during the DDoS attack. *Average serve rate* is an average amount of responses generated by server as responses to requests generated by legitimate client during the DDoS attack; during attacks the legitimate requests are often dropped. In SYN flood attack, Average serve rate is represented by the round trip time (RTT) response (SYN+ACK) to client's legitimate SYN request; in HTTP Get flood attack, by web page view requesting from legitimate client; and in SMTP mail flood attack, by client's email resending.

As can be seen from Fig. 3, 4 and 5, all performed experiments led to real DDoS attacks.

The SYN flood attack belongs to one of the most common types of DDoS well-known for a decade. From this reason, the attack was tested as first. Moreover, the employed production server contains countermeasures called SYN cookies against the SYN flood attack. Currently, the SYN cookies defense mechanism is a standard feature of Linux operating system. The SYN cookies mechanism closes each SYN queue entry until it receives a subsequent ACK response from the client. In this way, the SYN queue is not filled up with attacking connections and the SYN queue entry is reconstructed only when the server receives legitimate client ACK response [22]. Thanks to this mechanism, the attack had no significant impact on the server and values of all metrics correspond to normal operation conditions, Fig. 3. On the other hand, the

experiment proved the functionality of the defense mechanism under such a heavy load.

The second performed DDoS attack had an overwhelming impact on the victim server. As Fig. 4 illustrates, the load of the server during 60 seconds of the HTTP Get flood attack reached the value of 18, the main memory usage increased 7-times. Additionally, serving of the client requests was unavailable after 19 seconds. It should be noted that although http services were unavailable, the server was not shut down because of the security limitation set on the Apache web server.

In the SMTP mail flood attack, an email was sent every 5 seconds from legitimate client, with only one email successfully delivered during the test period. Since the attack ended after 60 seconds, the attacked mail server could more gradually deliver undelivered emails from the mail queue in which it had more than 113,000 pending messages. Finally, only one more email was delivered with a long time delay of 3 hours 7 minutes 48 seconds. Since most mail programs repeatedly send the undelivered mail, the attack is more likely to slow down, not to completely disable the server. On the other hand, the CPU load of the server during 60 seconds of the attack exceeded the value of 15, the main memory usage increased 4-times, which is a high server load, Fig. 5. Moreover, at the end of attack, the SMTP server was often disconnected from the client and the exception SMTPServerDisconnected was appearing more and more often.

It should be noted that it is particularly easy to increase the number of connected machines into the testbed environment and so to increase the number of attacking bots; it depends entirely on the number of available computers on the tester's side. On the other hand, although orchestration increases the ease of maneuvering, it also makes it easier for potential attackers to control the entire system from one central node. If an administrator's Ansible account is compromised, the attacker receives direct access to all computers managed from that account. For this reason, it is necessary to require intensified protection to the Ansible management node and to prevent its compromising.

V. CONCLUSION

The main aim of the DDoS attack is to direct the enormous malicious traffic from multiple computers to victim server and network to flood it. Ansible environment was used as a basis to develop and deploy a DDoS testbed and the testbed was tested for scientific experiments and study purposes. Performing and coordinating of the attack is supported by orchestration as the main feature of the open source Ansible. Different DDoS attack scenarios were performed, which led to degradation of victim's services under the attacks.

Proposed environment requires no special hardware; however it supports properties such as easy installation and management scalability, distributed environment as well as system implementation. The results of performed experiments confirm that testing environment based on Ansible tool is a powerful and effective tool to conduct real-conditions DDoS attack and can serve researchers as well as network administrators to test and analyze the state of security of their networks. Future work lays in involving IoT devices into automated job processing to perform IoT DDoS attacks.

REFERENCES

- [1] Netscout Systems, Inc. "Cloud in the crosshairs, NETSCOUT's 14th Annual Worldwide Infrastructure Security Report", Available on: https://www.netscout.com/sites/default/files/2019-03/SECR_005_EN-1901%E2%80%93WISR.pdf
- [2] P. Masek, et al. "Unleashing Full Potential of Ansible Framework: University Labs Administration." In: The 22nd Conf. of Open Innovations Association (FRUCT). IEEE, Finland, 2018.
- [3] M. Šimon, L. Huraj. "DDoS testbed based on peer-to-peer grid". IEEE Int. conference on Signal Processing, Communication, Power and Embedded System (SCOPE5). IEEE, 2016, pp. 1181-1186.
- [4] M. Šimon, L. Huraj, V. Siládi. "Analysis of Performance Bottleneck of P2P Grid Applications". In: J Appl Math Stat Inf., 9(2), 2013, pp. 5-11.
- [5] M. Šimon, L. Huraj, and M. Hošťovecký. "A Mobile Botnet Model Based on P2P Grid". In: Conf. on Creativity in Intelligent Technologies and Data Science. pp. 604-615. Springer, Cham, 2017.
- [6] S. Behal, and K. Kumar, "Trends in Validation of DDoS Research." In: Procedia Computer Science, 2016, 85: 7-15.
- [7] J. Dykstra. "Essential cybersecurity science: build, test, and evaluate secure systems." O'Reilly Media, Inc., 2015.
- [8] I. Halenar, M. Juhas, B. Juhasova, and D. Borkin. "Virtualization of Production Using Digital Twin Technology." 2019 20th International Carpathian Control Conference (ICCC). IEEE, 2019.
- [9] D. Schmidt, and S. M. Shalinie. "DDoS Testbed. An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks." Springer India, 2011. 115-129.
- [10] S. Behal, and K. Kumar, "Detection of DDoS attacks and flash events using novel information theory metrics," Computer Networks, 116, 96-110, 2017.
- [11] R.A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, P. "Survey and taxonomy of botnet research through life-cycle." *ACM Computing Surveys (CSUR)*, 45(4), 45. 2013.
- [12] I. Ullah, N. Khan, and H. A. Aboalsamh, "Survey on botnet: Its architecture, detection, prevention and mitigation." In 10th IEEE Int. Conf. on Networking, Sensing and Control, 2013, pp. 660-665.
- [13] S. Thakur, S.C. Gupta, N. Singh, and S. Geddam. "Mitigating and patching system vulnerabilities using ansible: A comparative study of various configuration management tools for iaas cloud." In Information Systems Design and Intelligent Applications (pp. 21-29). Springer, New Delhi, 2016.
- [14] J. O. Benson, J.J. Prevost, and P. Rad. "Survey of automated software deployment for computational and engineering research." 2016 Annual IEEE Systems Conference (SysCon). IEEE, 2016.
- [15] H. Beiknejad, H. Vahdat-Nejad, and H. Moodi. "P2P Botnet Detection Based on Traffic Behavior Analysis and Classification." Int. J. of Comp. & Info. Tech., (2018) 6(1): 01-12.
- [16] Pieplu, Rémi. "Ground Control Segment automated deployment and configuration with ANSIBLE and GIT." In: SpaceOps Conference. France, 2018.
- [17] L. Lu, Y. Feng, and K. Sakurai. "C&C session detection using random forest," Proceedings of the 11th Int. Conf. on Ubiquitous Information Management and Communication. ACM, 2017.
- [18] J. Berleur et al. "A Distributed Denial of Service Testbed." HCC9/CIP 2010, IFIP AICT 328, pp. 338-349, 2010.
- [19] Scapy 2.4.3.dev33 documentation <https://scapy.readthedocs.io/en/latest/>
- [20] ab - Apache HTTP server benchmarking tool, <http://httpd.apache.org/docs/2.0/programs/ab.html>
- [21] M. Haddadi, and R. Beghdad. "DoS-DDoS: Taxonomies of attacks, countermeasures, and wellknown defense mechanisms in cloud environment", EDPACS, 57:5, 1-26, 2018.
- [22] J.J. Echevarria, P. Garaizar, J. Legarda. "An experimental study on the applicability of SYN cookies to networked constrained devices". In: Software: Practice and Experience, 2018, 48.3: 740-749.

On the Carrier Suppressed Optical Modulation Methods for High Capacity DWDM Systems

1st Tomáš Huszaník

Department of Electronics and
Multimedia Communications,
Faculty of Electrical Engineering and
Informatics, Technical University of
Košice,
Košice, Slovakia
tomas.huszanik@tuke.sk

2nd Ján Turán

Department of Electronics and
Multimedia Communications,
Faculty of Electrical Engineering and
Informatics, Technical University of
Košice,
Košice, Slovakia
jan.turan@tuke.sk

3rd Ľuboš Ovseník

Department of Electronics and
Multimedia Communications,
Faculty of Electrical Engineering and
Informatics, Technical University of
Košice,
Košice, Slovakia
lubos.ovsenik@tuke.sk

Abstract—This paper deals with the optical carrier suppression (OCS) over optical fiber dense wavelength division multiplexing (DWDM) system considering advanced optical modulation methods, such as optical differential quadrature phase shift keying. The method of suppressing the carrier within the modulation process is the useful method to mitigate fiber nonlinear effects and thus reach better overall performance. The quantitative simulation results validate that the carrier suppression could improve the resistance of fiber nonlinear effects in the high capacity DWDM system.

Keywords—carrier suppression, DWDM, fiber optics

I. INTRODUCTION

The current generation of optical fiber networks relies mainly on the basic optical modulation techniques such as OOK (On-Off Keying) or OPM (Optical Phase Modulation). These modulation techniques are satisfactory for the majority of applications. However, with the ever-increasing demand of multimedia, high data rates and transmission distances these modulation techniques become inadequate. Due to the nature of the single-mode optical fiber, the optical nonlinear effects are dominant degradation mechanisms in high data rate and long-distance transmission [1].

From the study of nonlinear effects, it has been found that the generation of nonlinear effects SPM (Self-Phase Modulation), XPM (Cross-Phase Modulation) and FWM (Four-Wave Mixing) can be minimized if the transmitted signal has some special characteristics. A narrow bandwidth modulation format can increase spectral efficiency and chromatic dispersion resistance. Conversely, the modulation format with constant optical performance may be less prone to SPM and XPM. For today's demand, the optical differential quadrature phase shift keying is the most suitable optical modulation technique. Since single LiNb Mach-Zehnder modulator (MZM) is used to generate basic optical modulation such as OOK or 2-state OPM, dual-drive LiNb MZMs are popular and widely used today. The performance of dual-drive LiNb MZM can be increased by deploying optical carrier suppression (OCS). Fiber nonlinear effects are the main degradation mechanisms in optical fibers. The contribution of the phase modulation distortion and cross-phase distortion caused by self-phase modulation (SPM), cross-phase modulation (XPM) and four wave mixing (FWM) can be controlled by precise shaping of the spectral envelope of the transmitted signal. OCS is the interesting method that can be used to mitigate the negative impact of

fiber nonlinear effects, which limit the transmission distance and the maximum achievable data rate [2-6].

In this paper we present the simulation model of 16-channel DWDM system with the transmission rate of 100 Gbps per channel deploying optical DQPSK (Differential Quadrature Phase Shift Keying) modulation. The main aim is to evaluate the impact of fiber nonlinear effects using the advanced simulation environment OptiSystemTM. The OCS technique can be used to mitigate fiber nonlinear effects and comparative analysis is also provided.

II. CARRIER SUPPRESSED OPTICAL MODULATOR STRUCTURE

The conventional optical DQPSK modulator structure based on two arms LiNb MZM driven by in-phase and quadrature-phase signal components is shown on Fig. 1.

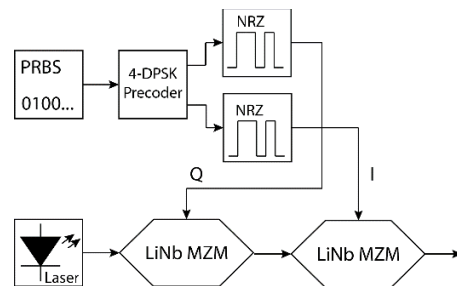


Fig. 1. Two LiNb MZM DQPSK modulator structure.

The modulator structure shown in Fig. 1 is commercially available in a LiNbO₃ integrated form [4]. There is a $\pi/2$ phase shift between I and Q signal component.

DQPSK modulator conveys information by establishing a phase of one symbol relative to the previous symbol while QPSK conveys the absolute phase of each symbol. This means, that standard DQPSK phase options are 0°, 90°, -90° and 180° or -180° while phase options of QPSK are 45°, 135°, 225° and 315°.

The induced phase difference between two arms of the modulator can be expressed by equation [3-5]:

$$\Delta\varphi_I(t) = \frac{u_I(t)}{v_\pi} \pi, \Delta\varphi_Q(t) = \frac{u_Q(t)}{v_\pi} \pi, \quad (1)$$

where $\Delta\varphi_I(t)$ and $\Delta\varphi_Q(t)$ are phase differences of I and Q signal components, $u_I(t)$ and $u_Q(t)$ are the applied voltage and

V_π is the driving amplitude. The transfer function of IQ modulator is [4]:

$$\frac{E_{out}(t)}{E_{in}(t)} = \frac{1}{2} \cos\left(\frac{\Delta\varphi_I(t)}{2}\right) + j \frac{1}{2} \cos\left(\frac{\Delta\varphi_Q(t)}{2}\right) \quad (2)$$

Using these two equations it is possible to calculate amplitude modulation $A_{IQM}(t)$ (3) and phase modulation $\varphi_{IQM}(t)$ (4) of IQ modulator [4].

$$A_{IQM}(t) = \left| \frac{E_{out}(t)}{E_{in}(t)} \right| = \frac{1}{2} \sqrt{\cos^2\left(\frac{u_I(t)}{V_\pi} \pi\right) + \cos^2\left(\frac{u_Q(t)}{V_\pi} \pi\right)}, \quad (3)$$

$$\varphi_{IQM}(t) = \arg \left[\cos\left(\frac{u_I(t)}{V_\pi} \pi\right), \cos\left(\frac{u_Q(t)}{V_\pi} \pi\right) \right] \quad (4)$$

The signal spectra of a 2 MZM based optical DQPSK modulator emitting on 193 THz is shown in Fig. 2.

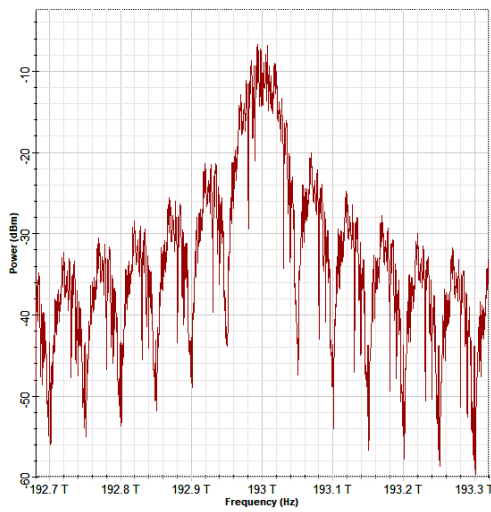


Fig. 2. Optical spectra of 2 MZMs based DQPSK modulated signal.

Fiber nonlinear effects, most notably SMP and XPM, are power sensitive. The negative contribution of these effects is well known and it is intensively studied. Another problem is a modulation instability that affects the performance of optical communication systems in which losses are periodically compensated using optical amplifier (such as DWDM systems). Simulation results showed that the main limiting factor is the employment of NRZ (Non-Return to Zero) format for data transmission [7-10].

The possible way to mitigate the nonlinear effects and achieve increased performance of DWDM system is to deploy carrier suppressed optical modulation. CSRZ-DQPSK (Carrier Suppressed Return to Zero DQPSK) is among the most promising optical modulation schemes for long haul and high data rate fiber optical systems providing increased tolerance to nonlinear effects and chromatic dispersion. Optical CSRZ-DQPSK modulator can be composed of 3 LiNb MZMs as shown in Fig. 3. To reduce the hardware complexity, it consists of a 4-DPSK precoder used to avoid recursive decoding at the receiver side. Precoder does the following mathematical operations:

$$I_i = \overline{(Q_{i-1} \oplus I_{i-1})(U_i \oplus I_{i-1})} + (Q_{i-1} \oplus I_{i-1}) \overline{(V_i \oplus I_{i-1})} \quad (5)$$

$$Q_i = \overline{(Q_{i-1} \oplus I_{i-1})(V_i \oplus I_{i-1})} + (Q_{i-1} \oplus I_{i-1}) \overline{(U_i \oplus I_{i-1})} \quad (6)$$

In the above formulas, \oplus denotes the logical operation of XOR (Exclusive OR). U_i and V_i are the input signals generated by PRBS. I_i and Q_i are the DQPSK coded output signals. Then, the logic operation $\overline{I_{i-1}}$, $\overline{Q_{i-1}}$ corresponds to the logic state 00 and the 180° phase difference. The logic operation Q_{i-1}, I_{i-1} corresponds to logic state 01 and 90° phase difference. 270° phase difference matches logic state 10 and Q_{i-1}, I_{i-1} logic operation. I_{i-1}, Q_{i-1} corresponds to logic state 11 and 0° phase difference [1-3].

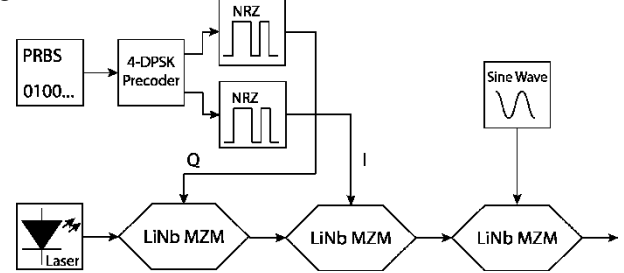


Fig. 3. 3 MZMs DQPSK modulator structure (CSRZ-DQPSK).

The first MZM is used to generate a phase shift of 0 or π of an in-phase signal component. The second MZM generates a quadrature phase shift of 0 and $\pi/2$ modulating the quadrature-phase signal component. So, the possible phase differences at the output of the second MZM are 0, $\pi/2$, $3\pi/2$ and π . To generate the carrier suppressed signal, the third MZM is used. It is used as a pulse carver. The pulse carving is realized by driving the MZM with a sinusoidal RF signal with the frequency of one quarter of a bit rate. The peak-to-peak amplitude is V_π . The signal spectra of a 3 MZMs based optical CSRZ-DQPSK modulator emitting on 193 THz is shown in Fig. 4 [11-13].

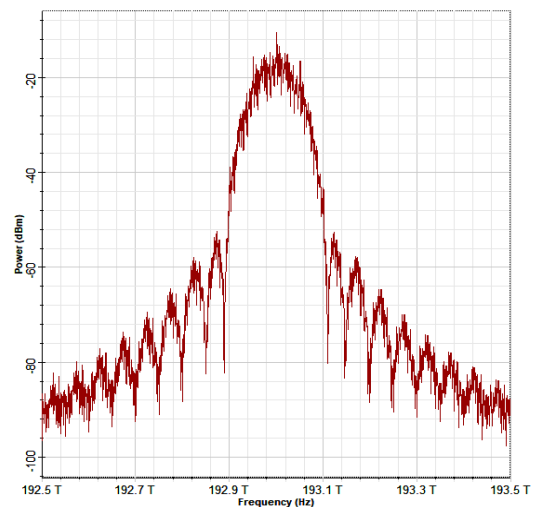


Fig. 4. Optical spectra of CSRZ-DQPSK modulated signal.

III. PROPOSED DWDM SIMULATION MODEL

The simulation model of 16-channel DWDM system is designed in OptiSystem™ simulation software.

The block scheme of the proposed simulation model is in Fig. 5. The simulation model consists of three subsections: transmitter section, optical distribution section and receiver section. The general simulation properties are: 100 Gbps bit rate, time window is 2.56e-09 s, sample rate 320 GHz and sequence length 256 bit. The number of wavelength channels is 16. The channel wavelengths are set according to the ITU-T G.694.1. The simulation is done in an optical C band (1530 – 1565 nm). Channels are then multiplexed by 16-channel arrayed waveguide grating (AWG) multiplexer with the attenuation of 5.5 dB.

Optical distribution path consists of 50 km of highly nonlinear single mode optical fiber (HNLF), two erbium doped fiber amplifiers (EDFA) and 10 km of dispersion compensation fiber (DCF). The optical distribution path is placed in the loop component. The number of loops is 15 (750 km). The advanced simulation model of SMF is used. It means, that both linear and nonlinear effect are taken into account. The attenuation is 0.25 dB/km, chromatic dispersion is 16 ps/km-nm², nonlinear refractive index is $n_2 = 2.6e-20$ m²/W and effective cross section area of the fiber is $A_{eff} = 80$ μm². Chromatic dispersion is fully compensated in DCF with dispersion set to -80 ps/km-nm². Channels are demultiplexed by AWG demultiplexer with 5.5 dB insertion loss.

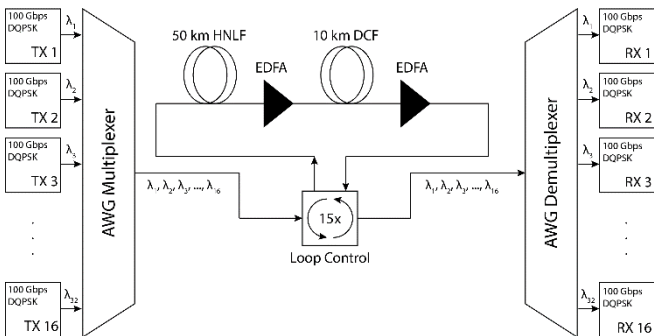


Fig. 5. 16-channel 100 Gbps DQPSK DWDM system.

Received signal is demodulated in a 90° optical hybrid shown in Fig. 6. Received signal is analyzed in electrical domain and from the received samples, the BER and Q-factor is calculated:

$$Q = \frac{I_1 - I_0}{\sigma_1 + \sigma_0}. \quad (7)$$

In the equation (7), I_1 is the mean value and σ_1 is the deviation of the pulse 1, I_0 is the mean value and σ_0 is the deviation of pulse 0. The bit error rate (BER) is calculated as follows:

$$BER = \frac{1}{2} \operatorname{erfc} \left(\frac{Q}{\sqrt{2}} \right) \approx \frac{1}{\sqrt{2\pi}Q} \exp \left(-\frac{Q^2}{2} \right). \quad (8)$$

The threshold for the BER value is 1.0e-10 and for Q-factor 6.3613. The performance of proposed DWDM system is not acceptable for BER higher than 1.0e-10.

Following simulation results are presented for received channel no. 5 (193.4 THz).

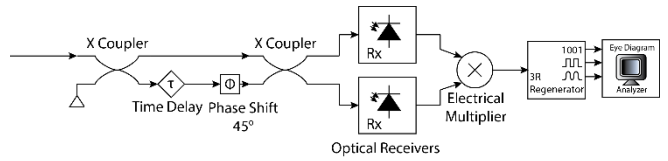


Fig. 6. Noncoherent DQPSK receiver.

IV. SIMULATION RESULTS

The performance of NRZ-DQPSK and CSRZ-DQPSK modulated 16-channel DWDM system with the data rate of 100 Gbps per channel was firstly analyzed over 15 optical segments. Each segment includes 50 km of HNLF, 10 km of DCF and two EDFAs. Fig. 7 shows the eye diagrams of received signals after 5 loops (250 km). The most important parameters of the eye diagram are the eye opening and the thickness of the lines. CSRZ-DQPSK modulated DWDM system shows better eye opening which indicates lower bit error rate (BER) and higher Q-factor. NRZ-DQPSK modulated DWDM system shows less opened eye diagram. Thicker the lines of the eye diagram, higher the bit rate. The thickness of the lines also indicates the OSNR (Optical Signal-to-Noise Ratio). In term of these results, the nonlinear effects can be mitigated by deployment of carrier suppressed optical modulation. Fig. 8 shows eye diagrams of the signals of the same DWDM system received after 550 km (11 loops).

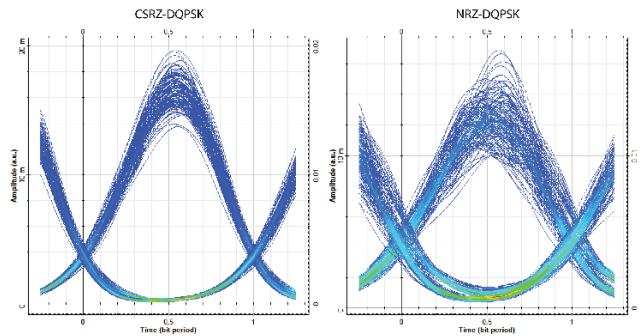


Fig. 7. Eye diagrams of received signals after 250 km.

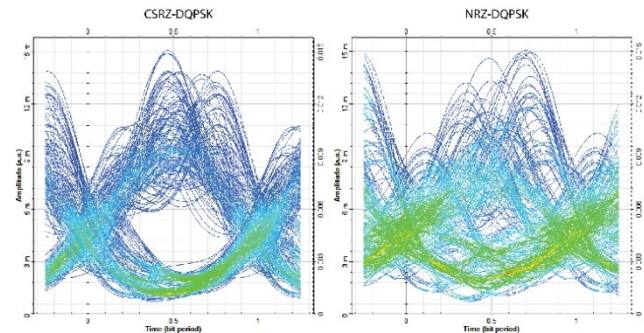


Fig. 8. Eye diagrams of received signals after 550 km.

The dependence between the length of the fiber optical transmission path and the modulation method used in DWDM is shown in Fig. 9.

In the next experiment, the effect of increased CW laser launch power on the creation of fiber nonlinear effects is analyzed. The transmission distance is set to 350 km (7 loops). The CW laser launch power is linearly swept from

0 dBm to 10 dBm. Again, the channel 5 is analyzed and BER is calculated.

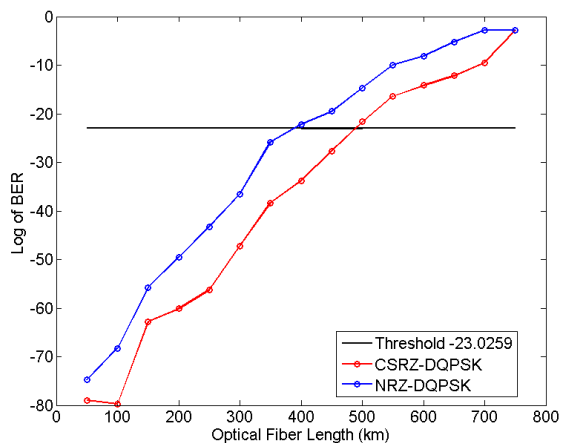


Fig. 9. Optical spectra of DQPSK modulated signal.

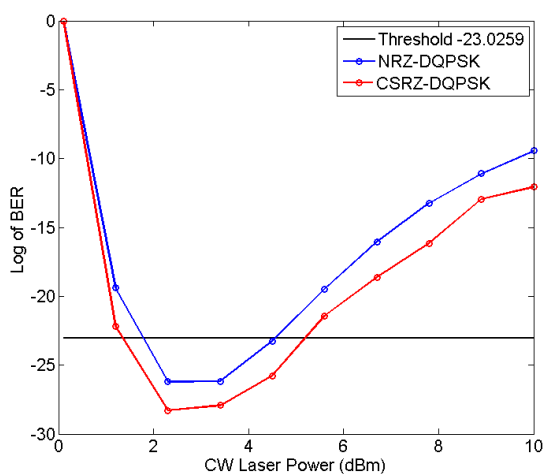


Fig. 10. Optical spectra of DQPSK modulated signal.

Based on the obtained results, the performance of 16-channel DQPSK modulated DWDM system can be improved using optical carrier suppression method. Using three MZMs DQPSK modulator structure, also known as CSRZ-DQPSK optical modulator, the negative contribution of fiber nonlinear effects can be mitigated. The maximum transmission distance for CSRZ-DQPSK modulated DWDM is 500 km. The BER is $3.96e-010$. The BER of NRZ-DQPSK modulated DWDM on the same distance is $4.25e-007$ which is not enough anymore. The launch power is also very important parameter due to the power dependence of SPM and XPM. CSRZ-DQPSK modulated DWDM system is more resistant to nonlinear effects due to the increased launch level. It means that we can manage the power in such DWDM more effectively. It also means that using the same launch level, CSRZ-DQPSK modulated DWDM system has the lower signal attenuation and thus transmission on the greater distance can be achieved.

V. CONCLUSION

In this paper we investigated 100 Gbps 16-channel DWDM system with an optical DQPSK modulation. We

provide the comparative analysis of the conventional 2 LiNb MZMs based NRZ-DQPSK optical modulator and 3 LiNb MZMs based CSRZ-DQPSK. Simulation results validate that carrier suppression can be useful way to mitigate the negative contribution of fiber nonlinear effects and achieve better performance of DWDM system.

ACKNOWLEDGMENT

This work was supported by following research grants: KEGA 023TUKE-4/2017, VEGA 1/0772/17 and APVV-17-0208 - Resilient mobile networks for content delivery.

REFERENCES

- [1] R. Ramaswami, K. N. Sivarajan, G. H. Sasaki. "Optical Networks," 2010. 928p. ISBN 978-0-12-374092-2.
- [2] J. M. Simmons, "Optical Network Design and Planning," 2014, 529 p. ISSN: 1935-3847, DOI: 10.1007/978-3-319-05227-4.
- [3] W. Ji and J. Chang, "Design of WDM-RoF-PON for wireless and wire-line access with source-free ONUs," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 2, pp. 127-133, February 2013. DOI: 10.1364/JOCN.5.000127.
- [4] M. Seimetz, "High-Order Modulation for Optical Fiber. Transmission," *Springer-Verlag Berlin Heidelberg*, 2009. ISBN 978-3-540-93770-8.
- [5] J. Tóth, L. Ovseník, J. Turán, L. Michaeli, M. Márton, "Classification prediction analysis of RSSI parameter in hard switching process for FSO/RF systems", *Measurement*, vol. 116, pp. 602-610, 2018, 10.1016/j.measurement.2017.11.044
- [6] V. Bobrovs, A. Udaļcovs, I. Trifonovs. "Evaluation of Nonlinear Effect Impact on Optical Signal Transmission over Combined WDM System," *Progress in Electromagnetic Research Symposium (PIERS 2013)*, Taiwan, Taipei, 25-28 March, 2013. Taipei: The Electromagnetics Academy, pp.303-307, 2013. ISBN 978-1-934142-24-0. ISSN 1559-9450.
- [7] P. Ivaniga, T. Ivaniga, J. Turán, L. Ovseník, M. Márton, D. Solus, J. Oravec, T. Huszanik. "The Influence of FWM with AWG Multiplexor in DWDM System," *Przeład Elektrotechniczny*, vol. 2018, no. 4, pp. 113-117, 2018. DOI:10.15199/48.2018.04.28.
- [8] T. Huszanik, J. Turán, L. Ovseník, "Comparative analysis of optical IQ modulation in four-channel DWDM system in the presence of fiber nonlinearities," *Proceedings of the 2018 19th International Carpathian Control Conference, ICC 2018*, p. 468-473. DOI:10.1109/CarpathianCC.2018.8399675.
- [9] W. Steingartner, M. A. M. Eldojali, D. Radakovic, J. Dostal, "Software support for course in semantics of programming languages," Paper presented at the *2017 IEEE 14th International Scientific Conference on Informatics, INFORMATICS 2017 - Proceedings*, , 2018-January, p. 359-364. DOI:10.1109/INFORMATICS.2017.8327275.
- [10] J. Kahn, K. P. Ho. "Spectral Efficiency Limits and Modulation/Detection Techniques for DWDM Systems," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 10, no. 2, p. 259-272 (2004). ISSN: 1558-4542. DOI: 10.1109/JSTQE.2004.826575.
- [11] R. Maharjan, I. Lavrinovica, A. Supe, and J. Porins, "Minimization of FWM effect in nonlinear optical fiber using variable channel spacing technique," *Advanced Wireless Optical Communication RTUWO 2016*, vol. 3, no. 3, pp. 1-4, 2016, DOI: 10.1109/RTUWO.2016.7821844.
- [12] A. Kaur, S. Dewra. "Comparative Analysis of Different Modulation Techniques in Coherent Optical Communication System," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, issue 8, p. 7193 – 7200, August 2015. ISSN: 2320-9801.
- [13] O. Kovac, P. Lukacs, I. Gladisova, "Textures classification based on DWT," Paper presented at the *2018 28th International Conference Radioelektronika, RADIOELEKTRONIKA 2018*, p. 1-5. DOI:10.1109/RADIOELEK.2018.8376379.

Interconnecting YAJCo with Xtext: Experience Report

Sergej Chodarev, Ján Halama

Department of Computers and Informatics, Technical University of Košice

Letná 9, 042 00 Košice, Slovakia

Email: sergej.chodarev@tuke.sk, jan.halama@student.tuke.sk

Abstract—YAJCo is a language development tool where the language is defined using a metamodel in the form of annotated Java classes. YAJCo can generate various tools for a defined language, including parser and pretty-printer. The goal of this work was to automatically generate support for the defined language in integrated development environments (IDEs). This goal is achieved by interconnecting YAJCo with Xtext — existing language workbench with support for multiple IDEs. The Xtext grammar is inferred automatically based on the defined metamodel, and subsequently used as input for generating. In this paper, we explain the concept of transforming metamodel to the Xtext grammar and generation of the IDE support. Attention is paid mainly to the differences between models. The solution is demonstrated on the created Mealy Machine Language and tested on the several examples implemented using the YAJCo tool in 4 chosen IDEs and editors — IntelliJ IDEA, Eclipse, Visual Studio Code and Atom.

I. INTRODUCTION

Computer languages, as well as the human ones, are always evolving – new paradigms and approaches which simplify software development are being introduced. Besides the general-purpose programming languages (GPL), domain-specific languages (DSL) used just within a specific domain have an important role in software engineering practice. They can be used in a broad spectrum of domains from computer education [1], to computer vision [2]. There are a lot of tools for creating computer languages relatively simple way — e.g. JastAdd [3], Rascal [4], MetaEdit [5], GME [6], Xtext [7], Spoofox [8], JetBrains MPS [9], etc.

Yet Another Java Compiler compiler (YAJCo) represents a language parser generator that allows defining language through meta-model in the form of Java classes. Relationships between these classes determine the abstract syntax of the language, while concrete syntax is defined by source code annotations [10]. The semantics of the language is expressed by methods of the metamodel's classes, or by separate classes that traverse the model. In addition to language parser, YAJCo also generates other tools, including pretty-printer and implementation of the Visitor design pattern that can greatly simplify the implementation of semantics, validation and other aspects of the language (see for example our recent case study [11]).

This paper focuses on the enhancement of the YAJCo by generating support for the newly-created language in chosen integrated development environments (IDE) and editors. YAJCo project tries to maximally utilize and integrate existing

technologies, so in this case, we also wanted to use an existing tool to provide this functionality. This goal has been achieved by interconnecting YAJCo and Xtext.

Xtext [7] provides excellent support for Eclipse IDE by generating a set of plugins for the defined language. Other IDEs and editors are supported by generating a server implementing the Language Server Protocol (LSP)¹ — an editor-independent interface for providing intelligent language-aware editing support, including code completion, symbol renaming, navigation, etc. So, in addition to Eclipse, we have focused on language support based on LSP in IntelliJ IDEA², Visual Studio Code³ and Atom⁴.

Also, the possibility of directly running the code written in the newly-created language has been implemented in the Eclipse plugin. This means that semantics of the language, defined by the YAJCo metamodel will be executed for the particular input sentence after it is transformed into the abstract syntax graph (ASG).

Moreover, Xtext generates an Ecore model [12] for every language out of its grammar. This Ecore model is instantiated when the input sentence is being parsed. This opens a perspective of integrating YAJCo with an ecosystem of Ecore-based tools including, for example, graphical model editors like Sirius [13].

II. DEFINITION OF LANGUAGE USING YAJCo AND XTEXT

In this section, we explain the way how languages are defined using YAJCo and Xtext. The differences between tools are shown in the example of Mealy Machine language — a DSL for defining Mealy finite-state machines. An example sentence of the language is presented in Listing 1. The semantics of the example language is defined as a translation into a configuration for the GraphViz tool used for generating a diagram of the machine. The diagram of the example sentence is presented in Fig. 1.

A. YAJCo

YAJCo is implemented as an annotation processor that transforms the input metamodel in the form of Java classes into the internal language model. Based on the created language

¹<https://microsoft.github.io/language-server-protocol/overview>

²<https://www.jetbrains.com/idea/>

³<https://code.visualstudio.com/>

⁴<https://atom.io/>

Listing 1. Example sentence of the Mealy Machine language

```

begin automaton
define input set: X0, X1;
define output set: Y0, Y1, Y2;
define state set: S0, S1, S2, S3, S4;
trans S0 to S1 when X0 produces Y2;
trans S0 to S2 when X1 produces Y0;
trans S1 to S1 when X0, X1 produces Y2;
trans S2 to S3 when X0 produces Y0;
trans S2 to S4 when X1 produces Y1;
trans S3 to S1 when X0 produces Y2;
trans S3 to S4 when X1 produces Y1;
trans S4 to S4 when X0, X1 produces Y2;
end automaton;

```

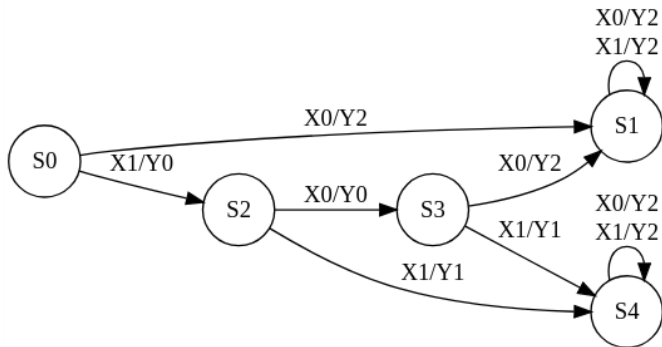


Fig. 1. Diagram of the example Mealy machine

model, the specific grammar is generated and used as an input for the particular parser generator (e.g. JavaCC, Beaver, ANTLR). The generated parser can parse a sentence of the language and produce a model instance.

YAJCo, as mentioned in the introduction, allows defining a new language using a well-known object-oriented way. The first step to define a language using YAJCo is to specify its abstract syntax (metamodel). The metamodel is expressed using Java classes and relationships between them, where each class represents a concept of the language. The generalization of concepts is expressed directly by class inheritance, while associations are represented by the parameters of class constructors. Order of the parameters together with their annotations define the concrete syntax of each concept.

Classes used to define the Mealy Machine language are shown in the class diagram in Fig. 2. The *Automaton* contains a set of inputs, outputs, and states and a collection of transitions. Transitions contain references to the inputs, outputs and states defined earlier. Class *InputReference* wraps a reference to the *Input* because currently, YAJCo does not support references with cardinality greater than 1.

Concrete syntax, which determines the exact valid notation of the sentence in a new language is specified by the several types of annotations. For example, in Listing 2 are presented fragments of classes *StateSet* and *State*. You can see the definition of terminal symbols using annotations `@Before`, `@After` and `@Token`. The delimiter between the objects in

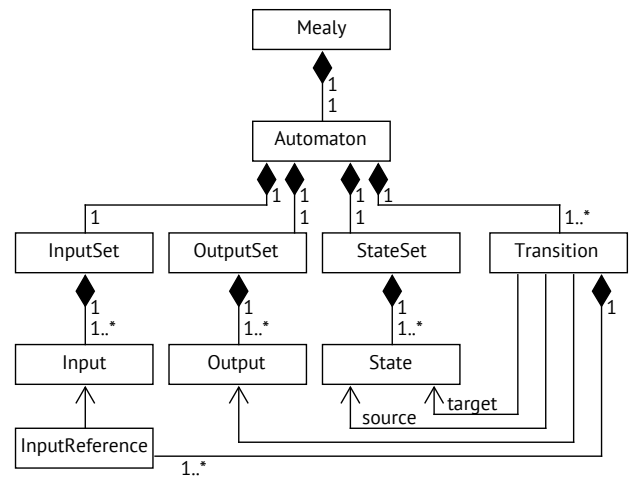


Fig. 2. Class diagram of the Mealy Machine Language metamodel

Listing 2. Fragments of Mealy Machine metamodel definition using YAJCo

```

public class StateSet {
    private State[] states;

    @Before("define_state_set:")
    @After(";")
    public StateSet(
        @Separator(",")
        @Range(minOccurs = 1)
        State[] states) {
        this.states = states;
    }
    ...
}

public class State {
    @Identifier(unique = "../State")
    private final String name;

    public State(@Token("name") String name) {
        this.name = name;
    }
    ...
}

```

a list, array or set is specified by the annotation `@Separator`, while their count can be defined by the annotation `@Range`. All the annotations are described in more detail by Porubán et al. [10].

YAJCo includes automatic reference resolving, so references between the classes are expressed by the annotation `@References`. In that case, instead of a full definition of the referenced concept, only its identifier is expected. A member variable, which is used as an identifier of the particular class is marked with annotation `@Identifier`.

The result of the parsing is a usual object-oriented model — an instance of the metamodel. Therefore it is possible to define its semantics in any appropriate way. For example as methods of metamodel's classes, or in a separate class implementing the Visitor pattern. In our example, semantic methods were de-

Listing 3. Xtext grammar of the Mealy Machine Language

```

grammar org.mealy.Mealy with org.eclipse.xtext.common.Terminals
generate mealyGrammar "http://www.example.org/mealy/MealyGrammar"

Mealy:
  "begin" name=NAME_TERMINAL automaton=Automaton "end" endName=NAME_TERMINAL ";" ;
Automaton:
  inputSet=InputSet outputSet=OutputSet stateSet=StateSet transition+=Transition+;
StateSet:
  "define state set:" states+=State ("," states+=State)* ";" ;
State:
  name=NAME_TERMINAL;
OutputSet:
  "define output set:" outputs+=Output ("," outputs+=Output)* ";" ;
Output:
  name=NAME_TERMINAL;
InputSet:
  "define input set:" inputs+=Input ("," inputs+=Input)* ";" ;
Input:
  name=NAME_TERMINAL;
Transition:
  "trans" initState=[State|NAME_TERMINAL] "to" finalState=[State|NAME_TERMINAL] "when"
  usages+=InputUsage ("," usages+=InputUsage)* "produces" output=[Output|NAME_TERMINAL] ";" ;
InputUsage:
  input=[Input|NAME_TERMINAL] ;

@Override
terminal WS:
  (' ' | '\t' | '\n' | '\r')+ ;
terminal NAME_TERMINAL:
  ('a'..'z' | 'A'..'Z') ('a'..'z' | 'A'..'Z' | '0'..'9')* ;

```

defined in classes representing language concepts and their goal is to generate a specification for the Graphviz for visualizing the machine.

B. Xtext

Creating the language using the Xtext follows the more traditional approach — it starts with defining the concrete syntax of the language. The syntax specification is augmented with annotations specifying its mapping to abstract syntax. After specifying the syntax, the semantics of the language can be defined using translation or interpretation.

In Xtext, the concrete syntax is expressed using a variant of the Extended Backus-Naur Form (EBNF). The concrete grammar of the Mealy Machine language is shown in Listing 3.

The first line of the grammar declares its name and optionally the name of mixin grammar (after `with` keyword), for example, a standard definition of common terminals (STRING, INT, WS, etc.). The second line defines the name and uniform resource identifier (URI) of the EPackage, which will contain Ecore model classes generated by Xtext.

The first rule of the grammar is the main one. In addition to usual EBNF operators for specifying alternatives, optionality, and repetition, Xtext requires to define the mapping of the concrete grammar to the abstract grammar. By default, each grammar rule corresponds to the language concept (metamodel class) named after the non-terminal on the left-hand side of the

rule. Non-terminals and literal values on the right-hand side of the rule are assigned to attributes of the concepts using the “=” operator. Multi-element objects like list, set or array are marked with special assignment operator “+=”.

Xtext also allows specifying referencing between the concepts using the square brackets. Inside the brackets is the name of the referenced concept and a terminal symbol representing the expected form of the reference, delimited by symbol “|”.

Terminal rules use a variant of regular expressions with syntax specific for the Xtext. The definition of the Xtext grammar is described in detail in [14].

Xtext supports implementing the semantics of a language as an interpreter and as a code generator (translator). In both cases, it is needed to implement a class for this purpose which contains methods for recursively walking through the model and defining the semantics for each model node. The interpreter or code generator is then bound to the particular action in the IDE (e.g. click on a context menu item, hover on language sentence, etc.). In the case of Mealy Machine DSL, the interpreter is triggered by clicking on context menu item *Interpret code* which prints the final GraphViz configuration to the IDE console.

III. INTEGRATION OF YAJCO AND XTEXT

We have utilized the modular nature of the YAJCo, where each generator is implemented as a separate module. To integrate it with Xtext we have developed a new module that

can be used alongside the usual parser generator modules. The result of the generator is a complete Xtext language project with a grammar and other artifacts that can be directly used to generate an Eclipse IDE plugin and a language server.

The process is divided into the following 3 steps:

- 1) generating Xtext language projects,
- 2) translating language definition into Xtext grammar,
- 3) generating and customizing other Xtext artifacts.

Support for Visual Studio Code and Atom requires, in addition to generated languages server, implementation of an extension that connects the server with the IDE or editor. We have prepared templates that can be used to create these extensions.

A. Generating Xtext language projects

The module uses the `CliProjectsCreator` class from the Xtext project to generate a complete Xtext language project. The resulting project consists of several modules (language definition, Eclipse plugin, language server) and uses Apache Maven⁵ as a build automation tool.

Xtext projects are generated automatically according to the configuration, which is partially defined by the user when creating the YAJCo metamodel (the whole name of the new language and the file extension) and partially by the YAJCo itself.

B. Translating language definition into Xtext grammar

Our translator uses YAJCo internal language model as an input and produces an Xtext grammar definition. It builds a custom Xtext language model as a modified version of the YAJCo language model. The differences between language definitions in YAJCo and Xtext are not only in the notation (annotated Java classes versus a variant of EBNF) but also in a fact that Xtext requires grammar definition to be tailored for the LL(*) parsing method. For this reason, an Xtext language model is used to store transformed grammar rules and write them into the final grammar definition file.

The basic structure of the YAJCo metamodel and Xtext grammar were described in Section II. This section describes the main differences between them in greater detail and shows the approach for transformation.

The grammar begins with a declaration header which includes the language name and its basic properties. It continues with non-terminal rules.

The first rule of the grammar is the main one – it is the beginning of the grammar. The main rule is in the YAJCo meta-model defined by attribute `mainNode` of the annotation `@Parser`. Other rules are written down to final grammar in a specific order according to their properties.

Concepts in the YAJCo model explicitly contain a lot of information that is expressed implicitly in Xtext. The most notable example is operator associativity and priority in expressions. In Xtext grammar, the associativity of the operator is expressed by the specific form of the rule, while priority is

given by the order of the operator in a chain of recursive rules. Therefore, grammar rules for operators need to be modified to encode the priority and associativity information in a way suitable for Xtext. Concepts with the same priority level are merged into super-rules and these rules are modified to include references to the next priority level. Modified rules should also contain a specification for model nodes creation that would ensure the correct form of the resulted ASG despite the modified grammar rules.

In the YAJCo model, it is possible to have multiple notations of the same rule which starts with the same token. In the context of Xtext, it is not possible without using backtracking, because of the ambiguity of such notation. This problem has been solved by the transformation of the list of notations to the list of trees, where each tree in the list represents separate notation, and merging the trees that have a common sequence of symbols in the beginning.

For example, concrete syntax of the rule `Condition` from the Imperative language example⁶ has two alternatives:

```
Condition:
  "IF" LPAR expression RPAR trueStatement |
  "IF" LPAR expression RPAR trueStatement
    "ELSE" falseStatement
```

When optimizing the notations corresponding trees are merged into one tree sharing the first five elements. The resulting tree is converted into the EBNF rule with the optional ELSE clause:

```
Condition:
  "IF" LPAR expression RPAR trueStatement
    ("ELSE" falseStatement)?
```

In addition to this, all the concepts that do not define concrete syntax (abstract concepts) are turned into alternatives listing all concepts that inherit from them.

After the non-terminal rules, terminal rules are written down to the grammar. These rules have the same order, as defined in the YAJCo meta-model. Tokens defined as `Skips` are transformed into the specific terminal rule — `WS`. This terminal rule represents white spaces, which are ignored in the created language and have no effect on the final ASG or semantics.

Terminal rules in YAJCo are defined using Java regular expressions. Xtext grammar, however, uses a different form of regular expressions. Therefore the own compiler has been implemented as a part of the integration of YAJCo and Xtext, which can translate a subset of Java regular expressions into Xtext form of the same expressions.

C. Generating qualified name provider

In addition to the grammar, a qualified name provider is generated for some concepts of the language. Its purpose is to define identifiers for the classes in the created Ecore model that are used in references. By default, Xtext uses the `name` attribute as an identifier, while YAJCo expects that a specific attribute would be marked using the `@Identifier`

⁵<https://maven.apache.org/>

⁶<https://github.com/kpi-tuke/yajco-examples/tree/master/yajco-example-imperativeLang>

annotation. This means that in cases, where the identifier defined in the YAJCo model is different from “name”, we need to override the default implementation.

IV. EXECUTING SEMANTIC ACTIONS IN IDE

Created Eclipse plugin is enhanced by the possibility to run the code written in the new language — executing semantic actions of the language. This is done by selecting the *Run code* item in the context menu.

Because there are many ways to define semantics in YAJCo, it is needed to define a class implementing the specific interface with a single method. Parameters of this method contain the input sentence and a stream which enables to write errors, logs, or other useful output into the console in the Eclipse IDE.

In the Xtext, an input sentence is parsed into the Ecore model inferred from the grammar. Although, the Ecore model contains all the member variables of the classes from the original Java meta-model it does not contain its semantic methods since they cannot be inferred directly from the generated Xtext grammar.

To be able to run the written code we need to either transform created Ecore model including actual values of its variables into the YAJCo model where the semantics methods are present or use the parser generated by YAJCo to produce its model directly. Instantiation of the object model using YAJCo, however, requires the execution of constructors of metamodel classes and the choice of a constructor depends on the exact notation used in the sentence and not only on the values of model attributes. For this reason, the second approach was chosen.

To do so, we need to ensure the availability of the meta-model’s compiled classes in the runtime of the Eclipse plugin. This means to copy the JAR file containing meta-model to the Xtext project representing this plugin and to modify the *MANIFEST.MF* and *build.properties* files accordingly. Modifications of the mentioned files allow us to run the parser directly in the code of the Xtext project. Running of the parser is specified directly in the implementation of the mentioned Java interface, which is called whenever the *Run code* option is chosen.

V. RESULTS

In addition to implemented Mealy Machine language, the implementation of generating support was tested on the 8 examples, available in the official YAJCo repository⁷. These languages included mathematical expressions language, simple imperative programming language, JSON parser, and language for controlling robot Karel.

The most complex of these languages is the Imperative programming language defined by 45 concepts in YAJCo, including 28 concepts for different kinds of operators. Generated Xtext grammar for this language contained 30 non-terminal rules. The main reason for the lower number of rules in the resulting grammar is the merging of rules representing operators of the same priority levels.

⁷<https://github.com/kpi-tuke/yajco-examples>

For each language, we have verified the correctness of the generated Xtext grammar by brief manual checking and by generating IDE support based on the grammar.

In the second step, the integration with IDE was verified in the Eclipse IDE and several IDEs and editors supporting the LSP (IntelliJ IDEA, Visual Studio Code and Atom). We have used example sentences to manually check if that generated parser and IDE plugin correctly provide the following features:

- code validation according to grammar,
- code completion,
- navigation to declaration or usages of a symbol,
- renaming all occurrences of a symbol,
- highlighting of language keywords (only in Eclipse).

Extensions that used generated LSP server where missing syntax highlighting because this feature is not supported by the protocol. It is our goal to implement custom generation of syntax highlighting specifications for other editors. The most plausible solution seems to use TextMate format⁸ for highlighting definition that is widely supported by different editors.

The main problems we have found were the limitations of the LL(*) parser generator used by Xtext. The grammar does not allow direct or indirect left recursion and therefore requires left-factoring of the grammar that would preserve the shape of the abstract syntax. We have already implemented a special-purpose left-factoring algorithm for the expressions with infix operators, however, a more generic solution needs to be implemented.

The second problem was caused by different notations used to specify terminals. While Xtext uses a custom form of regular expressions, YAJCo allows full Java regular expressions. While the richer notation provides additional flexibility, it limits analysis and translation of the expressions into different notations or replacement of lexer implementation. In the future, we may consider limiting constructs supported in token definitions to some basic form of regular expressions.

YAJCo represents a lot of language features in the explicit form (expressions with operators and parentheses, separators in lists, etc.). Some features, however, are not explicitly defined. The main example is the specification of different types of terminal symbols that may require special handling in IDE, like white-space, comments, strings, etc.

VI. RELATED WORK

This work extends the current possibilities of the YAJCo tool by adding the automatic generation of IDE extensions to support defined language. While traditional parser generators didn’t provide this functionality, it has become a standard component of *language workbenches* [15]. IDE support is also an important advantage of internal DSLs that reuse syntax and all the infrastructure of the host language, including its IDE, as was shown in the work of Nosál’ et al. [16].

Implemented concept, moreover, represents the interconnection between three technological spaces — *object-oriented*

⁸https://macromates.com/manual/en/language_grammars

programming (OOP), grammars and modeling. The concept of technological spaces is covered by Kurtev et al. [17]. This was possible because the two technologies interconnected in our work already work as bridges between the technological spaces — YAJCo connects OOP with grammars and Xtext connects grammars with modeling space.

The work of Butting et al. [18] presents an opposite approach compared to our work — it translates grammars to accurate metamodels. The goal is to reuse tooling available for EMF-based languages for languages defined using EBNF grammar. One of their main contributions is that the resulting metamodel contains also constraints derived from the grammar, for example, the cardinality of relations. Our approach in this respect relies on Xtext that generates Ecore metamodels without these constraints.

Object-oriented programming is not the only technological space, that can be connected with the Ecore Model through Xtext. XMLText tool [19] translates XML Schema into Ecore metamodel and generates Xtext grammar for such language. One of our previous works [20] is using a similar approach for interconnecting XML Schema with YAJCo.

VII. CONCLUSION

In this paper, we have presented the implemented concept of interconnecting the YAJCo and Xtext. This interconnection enables YAJCo to generate support for newly-created languages in Eclipse IDE and all the IDEs and editors supporting Language Server Protocol. Moreover, the generated Eclipse plugin has been enhanced by the possibility to run the code, written in the language. We believe that implemented extension can increase the attractiveness of the YAJCo tool, since its new possibilities allow a user to use the created language comfortably from the many IDEs and editors, with the option to run written code in Eclipse IDE as well.

Another advantage of this concept is generating an Ecore model out of object-oriented metamodel. Generated models can be used as a basis for utilizing existing tooling that is part of the Eclipse Modelling Framework, including graphical model editors. Further usages can be the object of future research.

Other topics that are part of our planned future work include:

- universal left factoring implementation that would keep the proper structure of abstract syntax,
- direct mapping between Ecore metamodel and YAJCo metamodel that would retain additional constraints and properties defined in the YAJCo metamodel,
- support of syntax highlighting in IDEs based on LSP.

ACKNOWLEDGMENT

This work was supported by projects VEGA no. 1/0762/19: “Interactive pattern-driven language development” and FEI TUKE Grant no. FEI-2018-57 “Representation of object states in a program facilitating its comprehension”.

REFERENCES

- [1] E. Pietriková, J. Juhár, and J. Štátná, “Towards automated assessment in game-creative programming courses,” in *International Conference on Emerging eLearning Technologies and Applications (ICETA 2015)*. IEEE, nov 2015. doi: 10.1109/ICETA.2015.7558505. ISBN 978-1-4673-8534-3 pp. 1–6.
- [2] D. Połap, K. Kęsik, K. Książek, and M. Woźniak, “Obstacle detection as a safety alert in augmented reality models by the use of deep learning techniques,” *Sensors*, vol. 17, no. 12, p. 2803, 2017.
- [3] E. Söderberg and G. Hedin, “Building semantic editors using JastAdd: tool demonstration,” *LDTA*, p. 11, 2011.
- [4] P. Klint, T. van der Storm, and J. J. Vinju, “Rascal: A domain specific language for source code analysis and manipulation,” *SCAM*, p. 168–177, 2009.
- [5] K. Smolander et al., “Metaedit—a flexible graphical environment for methodology modelling,” 05 1991. doi: 10.1007/3-540-54059-8_85
- [6] A. Ledeczi et al., “The generic modeling environment,” *Intelligent Signal Processing*, 2001.
- [7] M. Eysholdt and H. Behrens, “Xtext - Implement your Language Faster than the Quick and Dirty way,” p. 307, oct 2010. doi: 10.1145/1869542.1869625
- [8] L. C. L. Kats and E. Visser, “The spoofax language workbench: Rules for declarative specification of languages and ides,” *OOPSLA*, 2010.
- [9] M. Voelter and V. Pech, “Language modularity with the mps language workbench,” *ICSE*, p. 1449–50, 2012.
- [10] J. Porubán, M. Forgáč, M. Sabo, and M. Běhálek, “Annotation based parser generator,” *Computer Science and Information Systems (ComSIS)*, vol. 7, no. 2, pp. 291–307, 2010. doi: 10.2298/CSIS1002291P
- [11] S. Chodarev and M. Bačíková, “Development of Oberon-0 using YAJCo,” in *International Scientific Conference on Informatics: Informatics'2017*. IEEE, nov 2017. doi: 10.1109/INFORMATICS.2017.8327233. ISBN 978-1-5386-0888-3 pp. 122–127. [Online]. Available: <http://ieeexplore.ieee.org/document/8327233/>
- [12] D. Steinberg et al., *EMF: Eclipse Modeling Framework*, 2nd ed. Addison-Wesley Professional, 2008. ISBN 978-0-321-33188-5
- [13] V. Vuyović, M. Maksimović, and B. Perišić, “Sirius: A rapid development of dsm graphical editor,” in *International Conference on Intelligent Engineering Systems: INES 2014*. IEEE, 2014, pp. 233–238.
- [14] Eclipse, “The grammar language,” online. [Online]. Available: https://www.eclipse.org/Xtext/documentation/301_grammarlanguage.html
- [15] S. Erdweg, T. van der Storm, M. Völter, L. Tratt, R. Bosman, W. R. Cook, A. Gerritsen, A. Hulshout, S. Kelly, A. Loh, G. Konat, P. J. Molina, M. Palatnik, R. Pohjonen, E. Schindler, K. Schindler, R. Solmi, V. Vergu, E. Visser, K. van der Vlist, G. Wachsmuth, and J. van der Woning, “Evaluating and comparing language workbenches,” *Computer Languages, Systems & Structures*, vol. 44, pp. 24–47, dec 2015. doi: 10.1016/j.cl.2015.08.007
- [16] M. Nosál, J. Porubán, and M. Sulír, “Customizing Host IDE for Non-programming Users of Pure Embedded DSLs: A Case Study,” *Computer Languages, Systems & Structures*, vol. 49, pp. 101–118, 2017. doi: 10.1016/j.cl.2017.04.003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1477842416301269>
- [17] I. Kurtev, J. Bézivin, and M. Aksit, “Technological Spaces: An Initial Appraisal,” in *International Symposium on Distributed Objects and Applications, DOA 2002*, 2002. [Online]. Available: <http://doc.utwente.nl/55814/>
- [18] A. Butting, N. Jansen, B. Rumpe, and A. Wortmann, “Translating grammars to accurate metamodels,” in *ACM SIGPLAN International Conference on Software Language Engineering: SLE 2018*. New York, New York, USA: ACM Press, 2018. doi: 10.1145/3276604.3276605. ISBN 9781450360296 pp. 174–186.
- [19] P. Neubauer, A. Bergmayr, T. Mayerhofer, J. Troya, and M. Wimmer, “XMLText: from XML schema to Xtext,” in *ACM SIGPLAN International Conference on Software Language Engineering: SLE'2015*. ACM, oct 2015. doi: 10.1145/2814251.2814267. ISBN 978-1-4503-3686-4 pp. 71–76.
- [20] S. Chodarev and J. Porubán, “Development of custom notation for XML-based language: A model-driven approach,” *Computer Science and Information Systems*, vol. 14, no. 3, 2017. doi: 10.2298/CSIS170116036C

Influence of CNN Input Modification for Red Blood Cells Trajectory Prediction in Blood Flow

Michal Chovanec
Tachyum, s.r.o.,
Bratislava, Slovakia
Email: michal.nand@gmail.com

Hynek Bachratý, Katarína Jasenčáková and Katarína Bachratá
University of Žilina,
Faculty of Management Science and Informatics,
Department of Software Technology,
Žilina, Slovakia
Email: hynek.bachraty@fri.uniza.sk,
katarina.jasencakova@fri.uniza.sk, katarina.bachrata@fri.uniza.sk
<http://cellinfluid.fri.uniza.sk>

Abstract—Computer simulations of blood flow in microfluidic devices help for their development. These simulations are limited by their computational complexity. A possible solution to this problem is to analyze the simulation output data using machine learning methods. We use convolutional neural networks (CNN) for red blood cells (RBCs) trajectory prediction, which is very important for blood flow modeling. In this paper, we study how miscellaneous modifications of a CNN input affect the results from the learning experiments. All performed neural network experiments have sufficient accuracy. We evaluated, which CNN input parameters and their values have a significant impact on experiments accuracy.

I. INTRODUCTION

A. Computer simulations of blood flow

The flow of blood or other fluids in microfluidic devices is currently extensively studied in the field of biology and medicine, see [9], [11]. However, testing of these devices is often costly and technologically demanding. Therefore, we use a simulation model to optimize these devices. This model is described for example in [3], [7], [8]. Our research group's long-term goal is to optimize the simulation model of the device used to capture circulating tumour cells (CTCs) and their separation from the other solid components of the blood. It can be used for early diagnosis and appropriate treatment of cancer from a blood sample.

In the computer simulation, blood is modelled as a suspension. It consists of blood plasma and solid components. Blood plasma forms 55% of blood volume and solid components form approx 44%, here RBCs, platelets and white blood cells are represented in a ratio 600 : 40 : 1. Therefore, the main part of the solid components forms RBCs, which makes the proper modelling of RBCs crucial for the valid simulation model. It includes the correct setup of elastic parameters of RBCs model, their interactions with liquid (plasma), channel surface and the other cells.

However, the simulation model is currently very complex in both time and calculation. This complexity arises from the channel shapes, the amount of the RBCs, duration of the simulation experiment as well as the size of the output data. Some experiments often run several days or even weeks. It

happens to be a limitation for extension of the simulation experiments with the slightly modified parameters, which is often needed for microfluidic device optimization. Although a simulation usually provides a large amount of the measured data, we usually need to use only a fraction of it to solve a given problem. Several analyses suggest [4], [5], [6] that the use of appropriate statistical methods to output data allows generally describe and characterize experiments. This inspired us to further studying of the output data using machine learning methods [5], [10].

As we mentioned before, the behaviour and especially the movement of RBCs is very important in simulation experiments. For this reason, we decided to predict firstly the trajectories of RBCs from simulation experiments. These predictions are to be used e.g. for:

- Prediction of trajectory prolongation from executed simulation experiment.
- Prediction of RBC's motion in particular position of a microfluidic channel, even in such positions, where no cell passed during the experiment.
- Prediction of RBCs trajectory from video records, useful for tracking algorithms.

In study [5], we designed the neural network model inspired by radial basis function networks and Kohonen's self-organized maps. The prediction error of the best network is around 5% in this work. Later [10], we focused on establishing and testing the suitability of several CNN architectures and their input types for RBCs trajectory prediction. We chose this approach since convolutional neural networks are shown to be useful to search for spatial features that are significant in our task, see [14]. The network learns to predict the velocity of the cell's centre from the time sequence of its previous positions. We concluded that the accuracy of the performed experiments is significantly better for one type of an input tensor denoted as the input tensor based on the spatial discretization of the simulation channel. Trajectory prediction error of these experiments is less than 2.5%. The way to determine the trajectory prediction error is described in the section VII-A. In this paper, we further study the input tensor

based on the spatial discretization. In 96 executed experiments, we investigate the effect of various modifications of the input on the quality of trajectory prediction. These modifications are simulation channel discretization size, the number of the cells included in the input, approximation of cell position and the length of the cell centre position's time sequence, from which the cell's velocity is predicted. By comparing the experiments with the various settings of these parameters, we determined what setups can be appropriate for predicting RBCs trajectories from similar simulation experiments.

II. TRAJECTORY PREDICTION USING CNN

In this section, we briefly describe the used learning model for trajectory prediction. For further details, see [10]. From the time sequence of a cell centre's positions, networks learn to predict its velocity vector in each channel position. After that, a predicted position of this cell centre is integrated from the predicted velocity in the following time step. By repeating this procedure, the network predicts the whole trajectory of each cell from its starting position, see Figure 1.

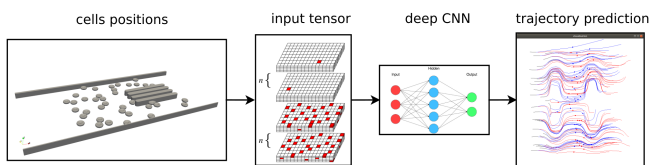


Fig. 1. Scheme of RBC trajectory prediction using CNN.

III. DATASET

Output data from the simulation experiment described below are the input data for the neural network. The motion of each cell is recorded during the simulation. We use coordinates of the cell centre position and velocity from the output data since our framework predicts velocity from positions. In the simulation data processed in this study, there are 9 057 records of every cell's motion.

Training and testing sets for the neural network comes from two simulations. These simulations vary only in the initial cells seeding. Thus, cells trajectories are different in these experiments. However, there should be similarities between these trajectories, since all other setups are the same. It includes the channel topology, elastic parameters of the cells and the parameters of liquid.

IV. SIMULATION EXPERIMENT DESIGN

The simulation experiment was performed using the open-source software ESPResSo. Our research group supplemented it with a module of elastic objects [8]. The fluid is modelled using the Lattice-Boltzmann method, see [1]. The model described in [15] is used to calibrate the cell membrane.

The simulation experiment is based on the biological experiment described in [16]. The experiment primarily investigated the study of the correlation between cells deformation and their velocity in the narrow slits formed by obstacles in the microfluidic channel. Internal dimensions of the channel are

$208\mu\text{m} \times 98\mu\text{m} \times 3.5\mu\text{m}$. It includes four obstacles depicted in Figure 2. The blood flow is from left to right, in the direction of the x coordinate. The simulation channel is periodic in this direction. It means that whenever the cell is leaving the channel, it is entering the same channel from the other side. The amount of the cells is 38, which corresponds to the haematocrit used in the laboratory experiment. Size of the modelled RBCs is $7,8\mu\text{m} \times 7,8\mu\text{m} \times 2,56\mu\text{m}$. At the beginning of the simulation experiments, all cells were in the left part of the simulation channel which is free from obstacles. The RBC model has 374 nodes. For the detailed description of the simulation experiment and its validation, see [13], [17].

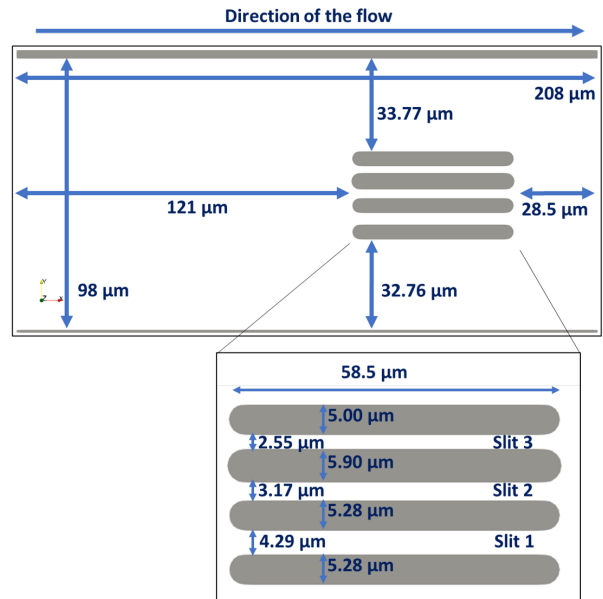


Fig. 2. Schema of the simulation channel with its dimensions

V. INPUT FOR THE NEURAL NETWORK AND ITS MODIFICATIONS

A. Description of the input tensor

The input tensor has three dimensions. It consists of n smaller 3D matrices $A_1^{(i)}$, $i = 1, \dots, n$ or of $2n$ smaller 3D matrices $A_1^{(i)}$, $A_d^{(i)}$, $i = 1, \dots, n$ depending on the parameter depth, i.e. amount of the cells included in the input. These matrices record position(s) of the cell(s) in the simulation channel at the particular time. The number n denoted the number of time steps when the cell centre positions are used for the velocity prediction in the next $(n + 1)^{th}$ time step. Note, that this time step is different from the simulation step. The first n 3D matrices describe the positions of the centre of the cell whose velocity is predicted. The other 3D matrices describe positions of all cell centres at the same n time steps. The individual time steps correspond to every 100^{th} output of the simulation. For this choice of the time step, the distance of the centre's positions of the same cell in two consecutive time steps is less than the

maximal possible radius of the cell. It means that the cell has a small intersection with itself in these consecutive time steps.

Now we describe how the position of the cell is recorded in the one 3D matrix $A_j^{(i)}$, $j \in \{1, d\}$. We have divided simulation channel into $disc_x \times disc_y \times disc_z$ cubic areas of equal dimensions by cuts parallel to the coordinates x, y and z . The 3D matrix $A_j^{(i)}$ corresponds to this discretization of the channel. The values of the matrix $A_j^{(i)}$ describe the occurrence of the cells centre in the appropriate area of the channel. The value of the $(x, y, z)^{th}$ element of $A_j^{(i)}$ is determined as follows:

Case point

$$A_j^{(i)}[x, y, z] = \begin{cases} 1, & \text{if there is a centre of some cell,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Case Gaussian

Let s^3 be the neighborhood of such element of $A_j^{(i)}$, whose value would be 1 in the case point. We replace all the values in this neighborhood with the values of Gaussian function e^{-l^2} , where l is a distance of area in the neighborhood from the area with the cell centre. The size of the neighborhood is $s = 5$ for all experiments studied here.

We make n 3D matrices, $A_1^{(1)}, \dots, A_1^{(n)}$, for n time steps. Each of them records the position of the centre of one cell. If the upper index of the matrix is bigger, then this matrix records the later position of the cell.

Case no depth

We do not want to incorporate all the cells to the input tensor. This tensor consists of matrices $A_1^{(n)}, \dots, A_1^{(1)}$ in that order. The lower index of the 3D matrices means, that these matrices record the position of only one cell.

Case depth

In this case, we incorporate all cells. We create another n 3D matrices $A_d^{(1)}, \dots, A_d^{(n)}$. Each of these matrices includes the information about every cell. Then the input tensor consists of 3D matrices $A_1^{(n)}, \dots, A_1^{(1)}, A_d^{(n)}, \dots, A_d^{(1)}$ in this order. Schemes of input tensors are in Figure 3. The dimensions of the input tensor are:

$$\begin{aligned} &disc_x \times disc_y \times disc_z * n, && \text{if } use_depth = false, \\ &disc_x \times disc_y \times disc_z * 2n, && \text{if } use_depth = true. \end{aligned}$$

B. Tested versions of miscellaneous input modifications

We tested 24 different neural network inputs with the following modifications of input parameters:

- 1) **channel discretization** size,
- 2) **depth** parameter is true if and only if all cells are incorporated in the input,
- 3) **position approximation** is point or Gaussian,
- 4) **time window** size is the number of time steps n used for velocity prediction.

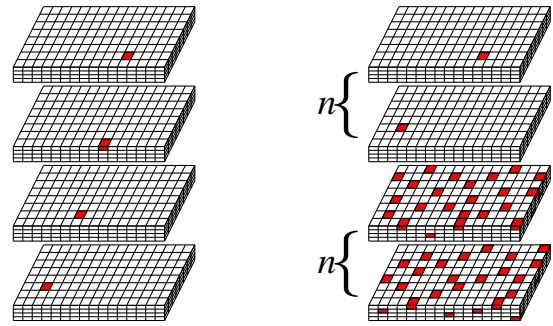


Fig. 3. Schemes of the inputs for $disc_x = 16$, $disc_y = 8$, $disc_z = 4$. Red areas represent positions of the cell centres. There are 3D matrices $A_1^{(i)}$ depicted on the left side, where the position of one cell centre is recorded. If we join these tensors together as is described above, we obtain the input tensor for the case no depth with $n = 4$ time steps. There is the input tensor for the case depth illustrated on the right side.

Table I represents the values of the input parameters used in CNN experiments. There are parameters of experiments studied in [10] in the first column. The parameters of the experiments used for this study are in the second column.

TABLE I
MODIFICATIONS OF THE CNN INPUT PARAMETERS

Experiments	Initial	Tested
channel discretization	$16 \times 16 \times 3$	$8 \times 8 \times 3$ $16 \times 16 \times 3$ $40 \times 20 \times 3$
Depth	no	yes no
Position approximation	Gaussian	Gaussian point
Time window	8	4 8

VI. NETWORKS ARCHITECTURES

We used four different CNN architectures in our experiments, see Table II. Experiments with these networks architectures and the input type described above are accurate enough as shown in [10]. We tested there these networks architectures only on one set of input parameters, see Table I. Therefore, we focus now on the comparison of the accuracy of experiments with different input parameters setups as well as the comparison of the experiments with the different networks architectures and the same settings of the input parameters. Besides using convolution or fully connected layers, these networks consist mainly of dense blocks, introduced in [12].

Each of the 24 inputs was used for 4 different CNN architectures. Hence, the number of experiments is 96.

A. Hyperparameters

We used following hyperparameters of neural networks architectures: Weights are initialized in the range *xavier*, bias is set to 0. Learning rate is 0.0002, $\lambda_1 = \lambda_2 = 0.000001$, dropout = 0.02 and minibatch size is 32.

Computations ran on CUDA, graphics card GeForce GTX

TABLE II
NETWORKS ARCHITECTURES

layer	net 4	net 5	net 6	net 7
0	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
1	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
2	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
3	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
4	conv 1x1x32	conv 1x1x16	conv 1x1x16	conv 1x1x32
5	fc 3	dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
6		dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
7		dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
8		dense conv 3x3x8	dense conv 3x3x8	dense conv 3x3x8
9		conv 1x1x32	conv 1x1x16	conv 1x1x32
10		fc 3	dense conv 3x3x8	dense conv 3x3x8
11			dense conv 3x3x8	dense conv 3x3x8
12			dense conv 3x3x8	dense conv 3x3x8
13			dense conv 3x3x8	dense conv 3x3x8
14			conv 1x1x32	conv 1x1x64
15			fc 3	fc 3

1080 Ti. Our algorithm is written in C++ and Python. The training phase lasts about 2 hours. Calculating the prediction of red blood cell trajectories on an already trained neural network takes approximately an hour and a half on the GTX 960 graphics card.

VII. ACCURACY OF RED BLOOD CELLS TRAJECTORY PREDICTION

Our goal is to find out for which type of network architecture and set of input parameters we obtain the best RBC trajectory prediction. We use so-called total trajectory error for this task. Since the prediction is supposed to replace the trajectory obtained by the simulation, we take into account not only its geometric shape but also its course in time.

The predicted trajectory of each cell is formed by the step-wise iteration of its position from the same starting position as the target trajectory. Cell position at each subsequent time step of the simulation is calculated from its current position and the corresponding offset in the direction of the velocity vector determined by the trained CNN.

A. Errors computing

Due to the needs of this study, the trajectory error is measured as the difference between the trajectory of the

simulation and the trajectory predicted by the neural network. This is determined for each cell as the difference between the desired and predicted position of its centre in the individual time steps of the simulation (Figure 4). Information about these trajectories is recorded in tensors. Y is the tensor of the required trajectories, \hat{Y} is the tensor of the predicted trajectories. The values of the tensor elements are coordinates of all cells centres in each time step. It is meaningful to compare trajectories from the moment when they are predicted. The prediction is from 800. time step for all experiments. For that, positions from this time step are recorded in tensors Y and \hat{Y} . The properties of the tensors Y and \hat{Y} are as follows:

- the dimension of the tensors is $depth \times height \times width$,
- index k specifies the cell, $k = 1, \dots, depth$, in experiments $k = 1, \dots, 38$
- index j determines the time step reduced by 800, $j = 1, \dots, height$, in experiments $j = 1, \dots, 8257$).
- index i identify the coordinate of the position, $i = 1, \dots, width$, in experiment $i = 1, 2, 3$.

As first we calculate the residuals:

$$E = Y - \hat{Y}. \quad (2)$$

Note that the E_{kji} , illustrated in Figure 4, denotes the $(k, j, i)^{th}$ residual of the matrix E .

We calculated different types of errors for individual coordi-

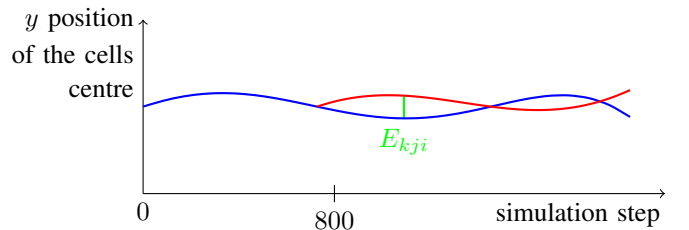


Fig. 4. Green line represents difference E_{kji} in coordinate y (with $i = 2$) in the simulation step $j > 800$ of the given cell k between target (blue) and predicted (red) trajectories.

nates as well as for the total error.

- 1) **Mean absolute error (MAE)** is computed for each coordinate as:

$$MAE_i = \frac{1}{38 \cdot 8257} \sum_{k=1}^{38} \sum_{j=1}^{8257} |E_{kji}|, \quad i = 1, 2, 3 \quad (3)$$

Calculation of total MAE is:

$$MAE = \frac{1}{3 \cdot 38 \cdot 8257} \sum_{k=1}^{38} \sum_{j=1}^{8257} \sum_{i=1}^3 |E_{kji}| \quad (4)$$

- 2) **Root mean square error (RMS)** is counted as the square root of the mean of the squares of the residues. For each coordinate is calculated as follows:

$$RMS_i = \sqrt{\frac{1}{38 \cdot 8257} \sum_{k=1}^{38} \sum_{j=1}^{8257} E_{kji}^2}, \quad i = 1, 2, 3 \quad (5)$$

total RMS is:

$$RMS = \sqrt{\frac{1}{3 \cdot 38 \cdot 8257} \sum_{k=1}^{38} \sum_{j=1}^{8257} \sum_{i=1}^3 E_{kji}^2} \quad (6)$$

- 3) For individual coordinates, **relative RMS error** is obtained by dividing the RMS error by the difference of the maximum and minimum tensor values in that coordinate. Thus, the error expresses the percentage of the total cell movement range. The total relative RMS error is obtained by dividing total RMS error by this difference in the x axis.

$$RMS_relative = \frac{RMS}{c_{max} - c_{min}} \cdot 100\% \quad (7)$$

where c is the corresponding coordinate x, y or z .

B. Evaluation of the accuracy of experiments

Neural network inputs are normalized in the range $(0, 1)$. However, errors are calculated in micrometres, i.e. in denormalized values. Since the simulation channel is narrowest in the direction of the axis z , normalization caused that the error is the most significant in this coordinate. Note that the channel height is $3,5\mu m$ and the RBC size is $7,8\mu m \times 7,8\mu m \times 2,56\mu m$. It means, the cell can move with very little rotation and a slight change of motion in the direction of the axis z . In this direction, the error in micrometres is very small. Therefore it can be neglected. Although we included it in the total error, it does not affect it very much. The relative RMS error in the coordinate x is in the range $\langle 2.66, 5.68 \rangle$ and for y axis from $\langle 3.63, 12.67 \rangle$. The movement in the direction of the x axis is mainly related to the flow rate. It is faster at the centre of the channel than at its edges and in channel narrowings. However, entry into the channel slits is associated with cell deformation, and then its rate is slowed. There are 3 slits in the simulation channel. During the simulations, through the narrowest slit did not pass any RBC. There were recorded 3 passages of the cells through the wider slit and 9 passages through the widest slit. Cells that passed through the second channel slit follow the desired trajectory shape, but position errors in time are large, see Figure 5. The reason is the mentioned different flow rate in this narrowing. Nevertheless, on average, these anomalies do not reflect and the total relative error is around 2.32%. For overall comparison of experiments, we used the total relative RMS error. In the following text, we will simply call it an error. This error is in the range $\langle 1.53, 3.26 \rangle$ for all 96 performed experiments. The two lowest errors have experiments with the input parameters listed in Table III.

In the following analysis, we describe the influence of input parameters and networks architectures on the accuracy of the trajectory prediction.

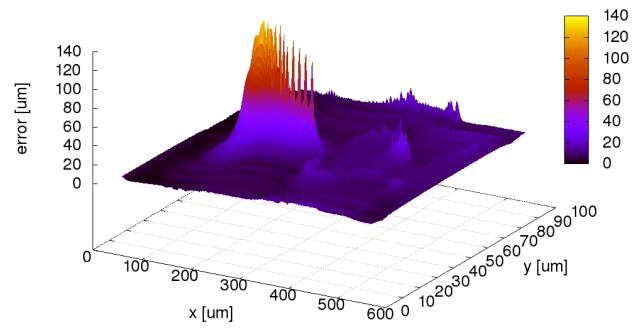


Fig. 5. RMS trajectory prediction error for all cells in the experiment.

TABLE III
INPUT PARAMETERS OF THE BEST EXPERIMENTS

total RMS		
relative error	1.53%	1.72%
channel discretization	$16 \times 16 \times 3$	$16 \times 16 \times 3$
Depth	no	no
Position approximation	point	point
Time window	4	8
CNN architecture	net 4	net 6

1) *The influence of simulation channel discretization on the trajectory prediction error:* Figure 6 shows the graph comparing error of experiments depending on the discretization of the simulation channel. In this and the following graphs, experiments are marked and ranked according to the values of other input parameters and network architecture. In this comparison, for more than 80% of experiments, the best accuracy is for discretization $16 \times 16 \times 3$. The average experiment error for this discretization is less than 2,05%. The experiments with discretization $8 \times 8 \times 3$ have an average error of more than 2.5% and with discretization $40 \times 20 \times 3$ around 2,4%. In all cases where the experiments with discretization $16 \times 16 \times 3$ do not have the smallest error, there is the time window size equal to 4 and the approximation of cell position is Gaussian. In this case, network architecture and parameter depth do not affect the accuracy of the experiments.

2) *Influence of parameter depth on trajectory prediction error:* The effect of the depth parameter shows the graph in Figure 7. In approximately 60% of the cases, there are more accurate those experiments in which not all cells

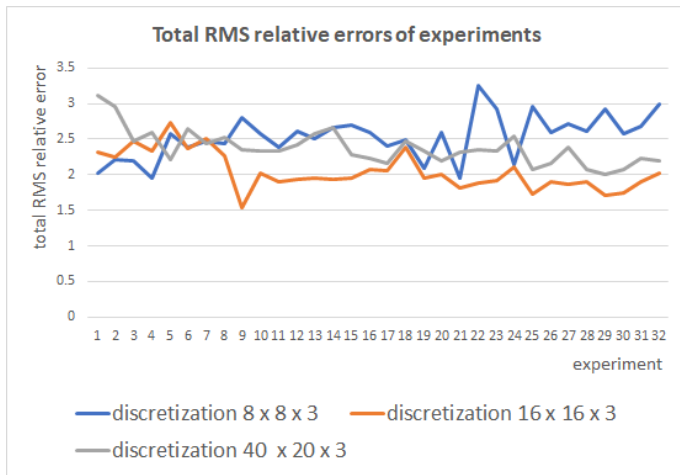


Fig. 6. Influence of channel discretization size on prediction error of experiments.

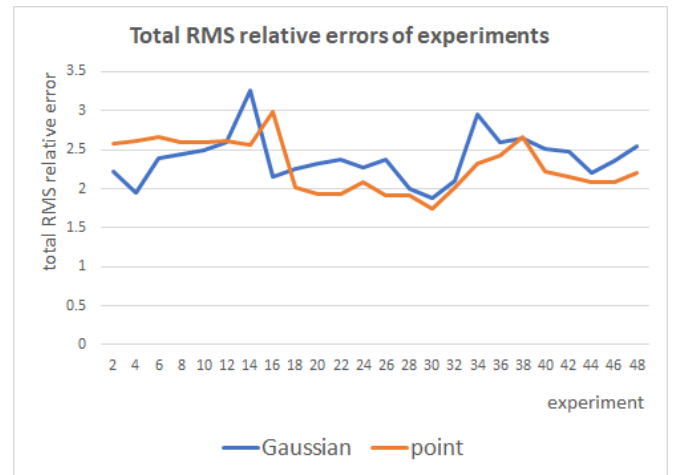


Fig. 8. Influence of position approximation of cells on prediction error of experiments.

were included. The average errors of experiments for both depth values are comparable and are approximately 2.3%. Therefore, we can conclude that this parameter has a minimal impact on the results of our experiments.

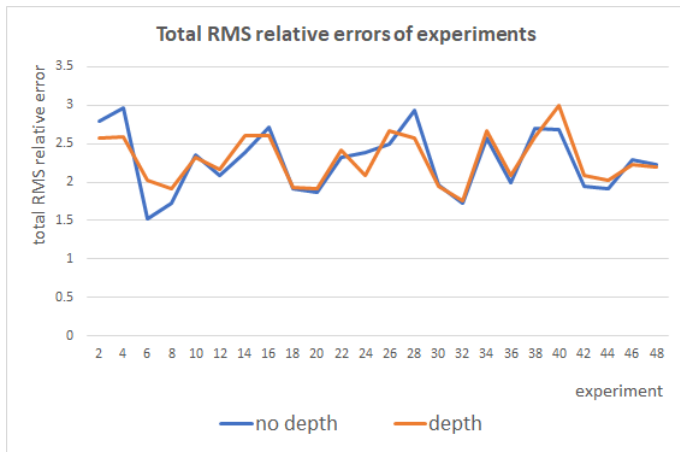


Fig. 7. Influence of input parameter depth on prediction error of experiments.

3) *Influence of cell position approximation on trajectory prediction error:* Figure 8 illustrates a graph that compares the error of experiments depending on the method of approximating the cell centre position. In the point case, a lower error is for two-thirds of the experiments. This approximation is better for all experiments with discretization $16 \times 16 \times 3$ and for more than 80% experiments with discretization $40 \times 20 \times 3$. The Gaussian approximation is more accurate for about 60% experiments from 1 to 16 with discretization $8 \times 8 \times 3$.

4) *Impact of time window size on trajectory prediction error:* The graph in Figure 9 depicts the error depending on the time window size. For more than 70% of the experiments,

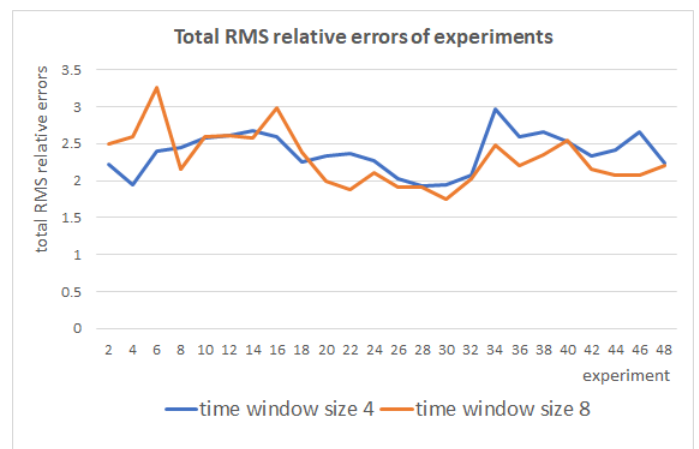


Fig. 9. Influence of time window size on prediction errors of experiments.

more accurate are those with the time window size equals 8. For discretization $16 \times 16 \times 3$, it is better to use the time window size 8 in almost 94% cases. Time window size 4 is more suitable only for experiments with discretization $8 \times 8 \times 3$.

5) *Impact of CNN architecture on trajectory prediction error:* For 24 combinations of input parameters, net 4 has the smallest error in six cases, net 5 in five, net 6 in eight and net 7 in five cases. Average experiment errors for these network architectures range from 2.27% to 2.35%. We can conclude, that for the chosen experiment configuration, network architecture has no significant impact on trajectory accuracy.

VIII. CONCLUSION

We found out that the total relative RMS error of the trajectory of the performed experiments is from the interval (1.53, 3.26). On average, this error is 2.32%. This is sufficient accuracy with respect to the purpose of cell trajectory prediction from the simulation experiment. Among the input parameters, the size of channel discretization has the

most significant impact on trajectory prediction accuracy. The most appropriate discretization is $16 \times 16 \times 3$. Among other input parameters, at least two-thirds of the experiments are better for the point approximation of cell position and time window size 8. For discretization $16 \times 16 \times 3$, experiments with these parameters are more precise in about 94% cases. Using the depth parameter does not have a significant impact on trajectory prediction accuracy. We can also say that the accuracy of experiments does not depend on the network architecture used in this study.

In the given simulation experiment, there were only a few passages of cells through the channel slits. Therefore, the network did not learn to predict the trajectory in these narrowings with sufficient precision. Nevertheless, the shape of the trajectories is predicted correctly.

In the future, we want to focus on predicting cell trajectories in simulations with a different topology of the channel and higher haematocrit. In such simulations should be more cell interactions that cause e.g. changes of cells rotation or their slope. In further research, we will also focus on predicting these characteristics using neural networks.

ACKNOWLEDGMENT

This work was supported by the Slovak Research and Development Agency (contract number APVV-15-0751) and by the Ministry of Education, Science, Research and Sport of the Slovak Republic (contract No. VEGA 1/0643/17).

REFERENCES

- [1] P. Ahlrichs, B. Dunweg, *Lattice-Boltzmann simulation of polymer-solvent systems*, Int. J. Modern Phys. C, 8, pp. 1429-1438, 1998,
- [2] A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Koovan, C. Holm, *ESPReso 3. Molecular dynamics software for coarse-grained models*, M. Griebel, M. Schweitzer (Eds.), Meshfree Methods for Partial Differential Equations VI, Lecture Notes in Computational Science and Engineering, vol. 89, pp. 1-23, 2013,
- [3] K. Bachratá and H. Bachratý, *On modeling blood flow in microfluidic devices*, ELEKTRO 2014: 10th International Conference, IEEE, ISBN 978-4799-3720-2, 2014, pp. 518-521, 2014
- [9] Y. Chen Y et al., *Rare cell isolation and analysis in microfluidics*, Lab Chip 14 (4), 626645, 2014
- [4] K. Bachratá, H. Bachratý and M. Slavík, *Statistics for comparison of simulations and experiments of flow of blood cells*, EPJ Web of Conferences, Vol. 143, art. no. 02002, 2017,
- [5] H. Bachratý, K. Bachratá, K., M. Chovanec, F. Kajánek, M. Smieková and M. Slavík, *Simulation of blood flow in microfluidic devices for analysing of video from real experiments*, Rojas, I., Ortuño, F. (eds.) Bioinformatics and Biomedical Engineering, 279-289, 2018,
- [6] H. Bachratý, K. Kovalčíková, K. Bachratá and M. Slavík, *Methods of exploring the red blood cells rotation during the simulations in devices with periodic topology*, 2017 International Conference on Information and Digital Technologies (IDT), Zilina, 36-46, 2017,
- [7] I. Cimrák, K. Bachratá, H. Bachratý, I. Jančígová, R. Tóthová, M. Bušík, M. Slavík and M. Gusenbauer, *Object-in-fluid framework in modeling of blood flow in microfluidic channels*, Communications, Scientific Letters of the University of Zilina, vol. 18/1a, pp. 13-20, 2016,
- [8] I. Cimrák, M. Gusenbauer and I. Jančígová, *An ESPReso implementation of elastic objects immersed in a fluid*, Computer Physics Communications, vol. 185, 900-907, 2014,
- [10] M. Chovanec, H. Bachratý, K. Bachratá and K. Jasenčáková, *Convolutional Neural Networks for Red Blood Cell Trajectory Prediction in Simulation of Blood Flow*, In Rojas I., Valenzuela O., Rojas F., Ortuño F. (eds) Bioinformatics and Biomedical Engineering. IWBBIO 2019. Lecture Notes in Computer Science, vol 11466. Springer, Cham, 284-296, 2019,
- [11] Q. Guo, S. P. Duffy, K. Matthews, E. Islamzada, Emel and H. Ma, *Deformability based Cell Sorting using Microfluidic Ratchets Enabling Phenotypic Separation of Leukocytes Directly from Whole Blood*, Scientific Reports 7, <https://doi.org/10.1038/s41598-017-06865-x>, 2017,
- [12] G. Huang, Z. Liu, L.V. Maaten and K.Q. Weinberger, *Densely Connected Convolutional Networks*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261-2269, 2017,
- [13] K. Kovalčíková, I. Cimrák, K. Bachratá, H. Bachratý, *Comparison of Numerical and Laboratory Experiment Examining Deformation of Red Blood Cell*, In Rojas I., Valenzuela O., Rojas F., Ortuño F. (eds) Bioinformatics and Biomedical Engineering. IWBBIO 2019. Lecture Notes in Computer Science, vol 11466. Springer, Cham, 75-86, 2019,
- [14] A. Krizhevsky, I. Sutskever, and G. Hinton, *Imagenet classification with deep convolutional neural networks* In NIPS, 2012,
- [15] R. Tóthová, I. Jančígová and M. Bušík, *Calibration of elastic coefficients for spring-network model of red blood cell*, International Conference on Information and Digital Technologies (IDT), pp. 376-380, 2015,
- [16] C. H. D. Tsai et al., *An on-chip RBC deformability checker significantly improves velocity-deformation correlation*, Micromachines, 7, 176, 2016,
- [17] K. Kovalčíková, H. Bachratý, K. Bachratá and K. Jasenčáková, *Influence of the Red Blood Cell Model on Characteristics of a Numerical Experiment* In Experimental Fluid Mechanics conference, Prague, 2018, in press.

Network motifs uncovering hidden characteristics of urban public transportation

Alexandru Iovanovici[†], Lilla Pellegrini^{*}, Anca-Maria Moscovici[§], Monica Leba[‡]

[†]Department of Computing and Information Technology, Politehnica University Timisoara

^{*}Doctoral School of University of Petrosani

[§]Department of Civil Engineering and Terrestrial Measurements, Politehnica University Timisoara

[‡] University of Petrosani, Department of Computer and Electrical Engineering

Email: [†]iovanalex@cs.upt.ro, ^{*}pellegrinililla@gmail.com, [†]anca.moscovici@upt.ro, [‡]MonicaLeba@upet.ro

Abstract—Transportation networks can be analysed using tools and techniques specific to complex networks analysis. We use curated data sets representing public transportation networks from 18 major cities around the world and by applying network motif discovery we extract significant characteristics which allow the urban planners of describing a specific scenario as "good".

Index Terms—Intelligent Transportation Systems, network motifs, complex networks, public transportation

I. INTRODUCTION

The public transportation system realizes the connection of the locality to the internal and international network and at the same time supports and influences the socio-economic evolution of one city. Accessibility, defined as the possibility of access to a desired destination, depends mainly on the extension and quality of the transport infrastructure and on the availability of services, being closely linked to that of connectivity, which is a usual term in network analysis.

Network analysis has become very popular in the last decades as it has proved to be immediately applicable in a large area of science. The Network approach has two benefits. One of them is that it simplifies and visualises the huge amount of data and the second benefit is that it has become very effective in picking out the most important elements and finding their most important interactions. Additionally, numerous techniques have been developed to discover the deeper topological structures of a network, such as community structure, core-periphery structure or small-world and scale-free properties [4]. These properties are usually the most common characteristic features of real-world complex networks.

Urban road network represents a spatial network as a result of the geographical features. The nodes and edges are fixed in space. The analysis and study of the topology structure could be the origin of the traffic state assessment and the optimisation of the traffic organization.

Numerous investigations of the transportation systems have been made in the last 20 years. The development of small-world networks and the appearance of modern graph theory lead to numerous studies about the topic of public transportation systems as complex networks. Many statistical characteristics have been published, for example the small-world property and scale-free distribution of various graph measures [1]. The public transportation is a network where the nodes are

represented by the bus stations and the edges connect successive stations. Beside the above-mentioned characteristics, in a network there can be found some small recurrent substructures, so called motifs. The study of the motifs has become a regular tool of complex network science, in order to accentuate the design basis of the structure of empirical networks.

II. NETWORK MOTIFS BACKGROUND

A network motif can be defined as a subgraph, usually with a small number of nodes, that appears significantly more frequently in the network than it does in a group of appropriately-chosen random graphs. The first mention of the motifs was done by Milo et al. who said that they are recurring, significant pattern of interconnection [11]. Milo et al. found them, among many others, in biochemical gene regulation networks and in the network of hyperlinks from the World Wide Web. They demonstrated that different sets of motifs are linked to diverse types of networks. Each motif could influence explicit functions, such as outlining universal network classes. Still, the existence and explanation of motifs in transportation networks has not been in the focus of researchers. The theoretic research on transportation system as complex networks characteristically concentrates on macroscopic structures like network diameter, or microscopic measures such as node centrality [11].

The importance of network motifs could be determined by the consideration that small subgraphs display special forms of links among network nodes and consequently they could have a regulatory or dynamic role. The frequency distribution of motifs in the network could be defined as a motif spectrum. This could be viewed as a pattern of the network structure and it permits to relate diverse networks and to create groups of networks with comparable significance profiles [4]. The analysis of motif spectra is an useful concept to unveil universal design principles underlying the structure of complex networks [6]. It is a tool to explain some properties of complex systems with relatively simple structures that led to an interest in network motifs in an expanding number of studies and across disciplines.

There are many ways to define a motif. Network motifs are the building blocks that profile the dynamic performance of a network or "patterns of interconnections occurring in

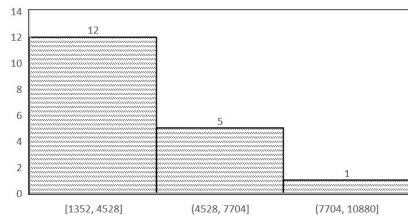


Fig. 1. Distribution of the city sizes seen as number of nodes in the corresponding network of public transport infrastructure show a balanced composition of the cities took under scrutiny in our study.

complex networks at numbers that are significantly higher than those in randomized networks” [9]. This definition could have been a little confusing mostly the random ensemble terms generated the confusion. Additionally, the terms “subgraph” and “motif” have been changed many times between them [8]. To be more specific, we choose to describe a motif as a group of topologically equivalent subgraphs of a network

III. DATA AND METHODOLOGY

Our investigation is targeting the public transportation network and consequently we have used one of the most extensive data-sets consisting of topology layouts and route traces available to the scientific community. The data-sets published by Kujala et al. in [6] consists of 25 cities form around the world for each of them having provided data regarding the layering of the network consisting of various “travel modes” as defined by the GTFS feed and for each of the layers we have access to a detailed description of it’s constituents (“bus” stops and routes represented as polylines). This level of details itself allows investigation in the realm of geo-information systems and urban geography but we were more geared towards the network properties which could be extracted form there. The authors used publicly available data fed trough General Transit Feed Specification (GTFS) but done the significant work of curating the data and preparing them for further study

We have took into account in our investigation cities of diverse size (Figure 1, on different continents and with a variety of local characteristics (separated by a river, on the seaside, having an old city center). Out of the bulk of data provided by the repository we have used the *networkNodes.csv* file which consists of public transportation stops which are going the function as network nodes and also the *network[Mode].csv* file which was selected only for bus and tram as discussed above.

A. Public transportation and complex networks

Data associated with each of the cities usually can be plotted as seen in figure 2 but beyond mere visual observation which can yield some insight into clustering of the station and routes there is no much data which can easily be obtained so we relied on transforming these data into a graph-like structure.

Before applying the motif finding techniques and extracting relevant patterns we have first inspected the PT networks we have selected using the tools of complex networks. Having

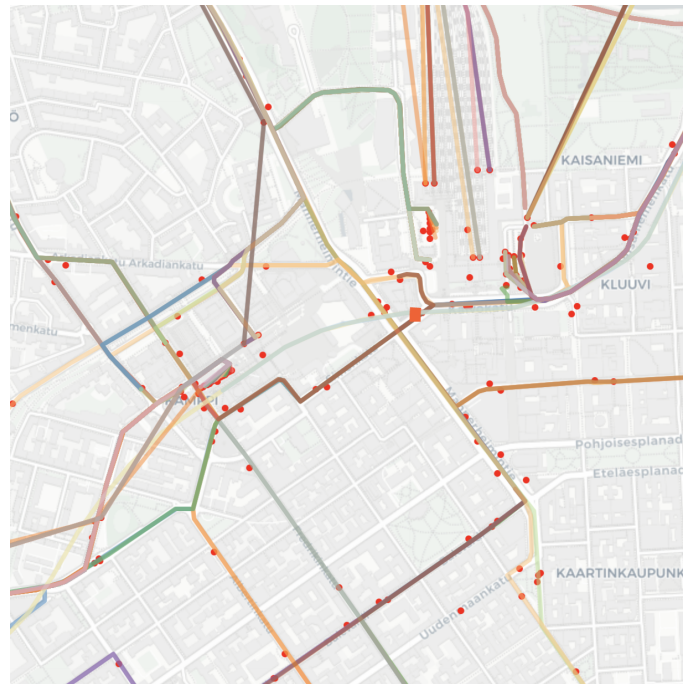


Fig. 2. Public transportation routes in center of Helsinki, red dots representing stops. Network is constructed by considering directed edges between each pair of consecutive stops along a route.

a large enough graph we hypothesise that there should be emergent properties which stem form the structure of the network. In figure 3 we show a network-centric representation of the transportation network for city of Athens in Greece. Each stop is associated to a node in the network while each directed edges are used to depict the travel route of the vehicle. The nodes are not placed based on their geographical positioning but the graph is rendered in Gephi using the Force Atlas 2 rendering algorithm which uses an approach stemmed for attraction an repulsion forces in physics to place the nodes at an equilibrium. Colors and sizing is going to be explained later on.

On the complex network side we measured classical metrics and their distributions. In Table I we show most important metrics computed for the cities we took under scrutiny. Number of nodes and edges provide a sense of the scale of the city and in Figure 1 one can perceive the almost equal melange between small, medium and large cities.

The other important metrics which describe a complex network from it’s topological point of view are depicted in Table I. The *average degree* is smaller then when we analyze the road network as a whole because in the case of transportation networks most nodes are represented by stops along a route and each node shall have exactly one directed connection (towards the next stop). The values are greater than one because in the case of some stops there are transfer stations or more than one line share the same stop and consequently there is going to be seen as “fork” in the network. Next important metric, which also can be considered a metric of

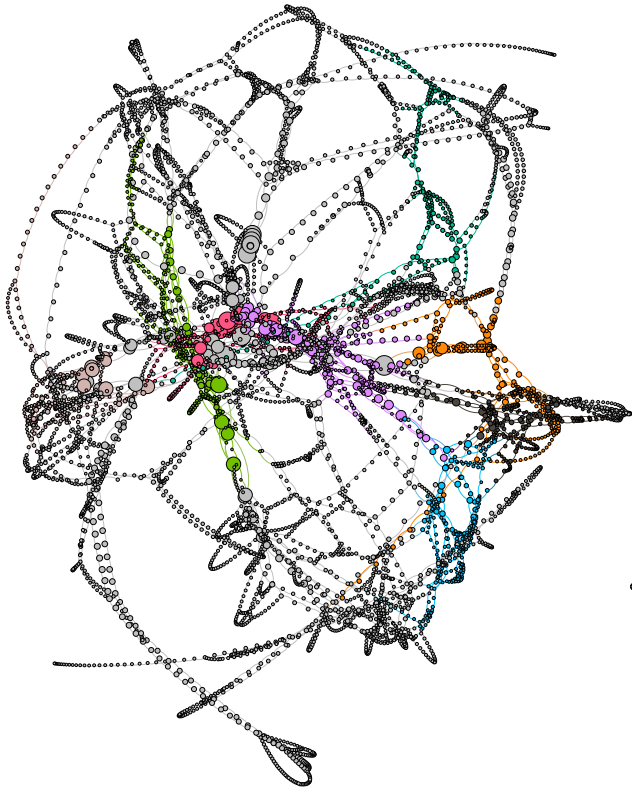


Fig. 3. Network representation of the public transportation in Athens, Greece. Nodes are representing stops and edges are directed towards the travel routes of the vehicles. Nodes are sized by the betweenness metric and colored by the community they are part of. Rendering layout is Force Atlas 2 as implemented in Gephi 0.8

size when we are dealing with physical cities is represented by *network diameter* which is simply the length (number of edges) of the longest of the shortest paths computer over each pair of nodes. One can perceive in this data-set that this is linearly dependant with the number of nodes, which is to be expected, but there are slight abnormalities such as the case of Paris where for 10880 transport stops the network diameter is merely 193, way lower than Detroit for example, which has 348 edges on its longest path while having only 4361 stops. It's worth noting that from a user's (commuter's) perspective the seemingly large numbers for the diameter are obtained from taking into account also the transit stations and not that a single line has more than 200 stops for example. The main reason for such disparities is stemming from the city topology and the way public transport is organized. Paris public transport relies heavily on "on-demand stops" where the commutes can press a button inside the vehicle the signal the driver for requesting to stop while other scenarios use fixed stops where the car actually stops even when there is nobody

to get in or out.

Going further we measured the *modularity* metric for each network. As the name says it's a measure of how well the network can be separated into smaller "chunks" later called communities. Our previous work on this subject [12] provided significant results related to road networks as a whole and cities having a good modular structure. Higher values for modularity signify a better and easier division while the lower value impose that the network does not have a clustered structure (such as artificial grid networks). It's worth noting that in this case, even if we deal with urban networks we don't take into account the geo-location of the nodes while relying only on the topological properties (network connectivity). Here we can perceive very high values for the modularity which is a strong indication of the community structure. This is consistent with both the literature [2, 5] and the empirical observations. Cities are organized and evolved around the geographical features and landmarks while the neighborhood are the micro-societal way or organizing living. The public transport evolved around these factors and most of the time communities in public transportation networks are consistent with neighborhoods while the connections between communities are represented by metro lines or other forms of "long-links".

Clustering coefficient is another metric we have determined for our data set. This is similar to the previous notion of modularity, while both being empirical metrics of measuring *clustering* i.e. the way nodes are grouped together. Clustering coefficient uses the number of triangles which can be found (some simple form of *motif* in relationship with the 2-paths (used by the modularity algorithm)). So, in our case low values for clustering coefficient are caused by the fact our networks have a lot of long 1-D paths along the travel route of our vehicle, consequently almost no triangles.

Putting all together we run the community detection algorithm, keeping each time the same constant value for the resolution parameter to 1.0. Due to algorithm's non-deterministic nature, for each set we have run the algorithm five times and taken the average value of communities rounded to the nearest integer. The lowest value of 22 communities is obtained for Venice with its very particular geographical features while cities with high values (Detroit-72, Paris-60) are either geographically big, or have a qualitatively good public transport infrastructure [7, 3], such as Dublin, Grenoble, Helsinki or Toulouse. For the example of Athens we have in figure 3 each node colored according to the community it was identified to be part of. One final aspect we took under scrutiny is the *betweenness* metric. This is a classical one in the realm of complex networks, is associated with a specific node and loosely is defined as the number of shortest paths we traverse the node when computing all the shortest paths between all the pairs of nodes in the network. Usually is perceives a centrality metrics, measuring the importance of the node to the networks, because empirically a node which is part of numerous paths should be an "important one" [10]. In figure 3 the nodes are sized according to their betweenness from largest to smallest.

B. Motif discovery and analysis

The main direction of our investigations is represented by *motif* analysis. Seen as subgraph or particular regular structure these represent structural patterns in the transportation network and can provide a static insight into the *good practices* of designing such systems. For all the data-sets we have used FanMod [13] for motif discovery. For each of these we have searched for motifs of 3 to 5 nodes. In figure 4 we show the most prevalent motifs of 3 to 5 nodes, directed, among the 18 cities we took under scrutiny. There, above each snapshot representation of the associated subgraph, the number indicates the unique ID associated with the FanMod database, for further referencing.

1) *3-node motifs*: First we took under consideration all the motifs which can be made out of three nodes. In this case ranking the findings among the 18 data sets, the top ones are depicted in figure 4. Even if they might look simplistic and obvious some considerations can be made based on them. The most prevalent is motif 12 which is simply any sequence of 3 stops along a route, followed closely by motif 38 which represent a forking in the transportation network where from a stop one can chose two close stops of another line. Motifs with id's 14 and 164 are representations of a similar situation in which along a route there is a stop where there is a merging with another line shared with the first one, but which run in opposite direction.

2) *4-node motifs*: Increasing the number of nodes we gain access into a more complex view of the network, allowing us to see interactions between two of more transportation lines. The leading pattern, with id 204 is simply caused by to line which share the same two stops (an obvious frequent situation). Going further we have id 2182 which depicts the scenario where two lines share two stops, but in-between them there is another stop, individual to each line. Next are 2076 and 2118 which similarly describe a situation into which along the route of a specific line, there is a portion where stops are shared with another "smaller" route. Finally motif 202 describe the topology of a *hub* where three lines share a single unique stop, without sharing adjacent stops (they are taking different routes)

3) *5-node motifs*: Going further we examined the structures consisting of 5 nodes, where usually one can find so called *hubs*. Leading motifs are those with id's 2133678 and 8948910 there the visually central node is the one share among there major lines (the edge is bidirectional, so the car goes both ways along the same path). On third place (id=1084606) we have an even bigger hub where all the three lines are converging into the same stop, but this is less prevalent mostly because of the limited possibilities of finding topographical conditions into various cities. One can find such structures into two largely distinct cases: big cities such as Paris and Detroit where there are numerous inter-modal stations (switching among various means of public transportation) or in smaller cities (such as Nantes, Rennes, Luxembourg or Venice) where the same concept of inter-modality exists, but usually between inner city routes and outer city ones. The fist most popular structure is

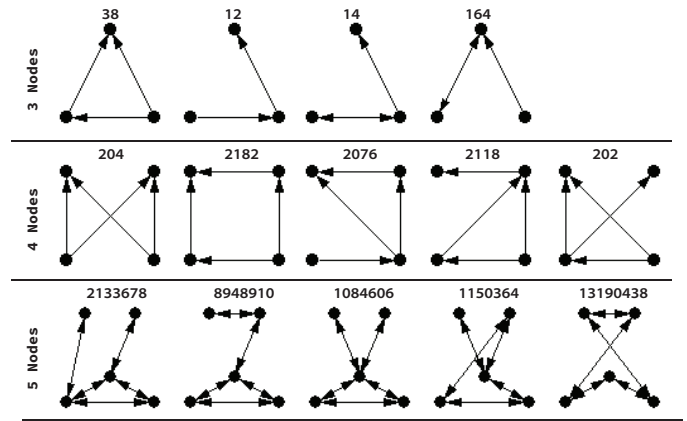


Fig. 4. Most prevalent directed network motifs of 3 to 5 nodes among the cities under study

the one with id 13190438, where there is a complete *square* between four stations shared among two lines which run their routes back-and-forth.

IV. DISCUSSION AND CONCLUSIONS

Public transportation networks provide an alternative to personal cars and other more polluting and expensive ways of commuting. Major cities involve a great deal of effort and budget for finding good solutions to reduce the congestion and transit times. Our investigation was geared towards applying methods and techniques form the complex networks to identify key characteristics which link the topology of public transportation networks among major cities. We used the concept of motif to describe patterns of stops and routes at a microscopic level and their distribution among the cities we have investigating. Ranging the node-size of the subgraph from 3 to 5 we have examined the distribution of motifs of that size, putting into evidence the actual connection to urban planning situations and the solutions provided in each case by the specific pattern/motif. When staying at low sizes (3 nodes) we don't get much insight beyond the obvious patterns but we can use this case for validating the method because the simple structures are much easier to be cross-referenced with city transportation maps by domain experts.

When we switch to 4 and 5 nodes the structures which unfold can truly go beyond simple observations on the map and uncover patterns among routes which share same stops. We can observe hub-like structure mostly when we examine the 5-node structures, where the top 5 most frequent motifs are all variations of the hub topology. Going beyond 5 nodes was at this moment unfeasible because the limited geo-visualization workflow did not allow us to cross-reference with the actual terrain situation for further explanations.

Going further we wish to extend this investigation with a quantitative examination of the influence of the motifs to the quality of the public transportation. Until now we have extracted the relevant data, and made empirical correlations between the prevalence of various motifs and their corresponding situations in the terrain, but being to quantify the impact

TABLE I
MAIN COMPLEX NETWORKS METRICS COMPUTED FOR THE CITIES UNDER STUDY

City	Nodes	Edges	Avg. Degree	Net Dia.	Modularity	Avg. clustering coef	#Comm, r=1
Athens	6658	7870	1,182	206	0,922	0,007	46
Belfast	1917	2181	1,138	146	0,903	0,005	31
Bordeaux	3212	3805	1,185	125	0,9	0,006	31
Detroit	5683	5948	1,047	348	0,953	0,005	72
Dublin	4361	5276	1,21	127	0,913	0,019	38
Grenoble	1383	1514	1,095	123	0,898	0,002	29
Helsinki	6642	8611	1,296	116	0,921	0,01	43
Lisbon	6949	8730	1,256	153	0,933	0,007	41
Luxembourg	1352	3127	2,313	53	0,872	0,073	28
Nantes	2208	2587	1,172	125	0,901	0,009	32
Palermo	2176	2561	1,177	138	0,893	0,004	27
Paris	10880	12604	1,158	193	0,93	0,003	60
Prague	4476	5911	1,321	109	0,906	0,018	37
Rennes	1378	1643	1,192	85	0,881	0,009	22
Rome	7500	9669	1,289	143	0,924	0,013	41
Toulouse	3237	3668	1,133	171	0,925	0,003	40
Turku	1850	2341	1,265	92	0,887	0,008	26
Venice	1733	2398	1,384	82	0,885	0,026	22

of each motif and link it to the overall quality of the service would allow designing better public transport infrastructure, avoiding "bad" patterns and favouring "good" ones.

REFERENCES

- [1] Marius Agasse-Duval and Steve Lawford. Subgraphs and motifs in a dynamic airline network. *arXiv preprint arXiv:1807.02585*, 2018.
- [2] Federico Botta and Charo I del Genio. Analysis of the communities of an urban mobile phone network. *PLoS one*, 12(3):e0174198, 2017.
- [3] Jorge Chica-Olmo, Héctor Gachs-Sánchez, and Carmen Lizarraga. Route effect on the perception of public transport services quality. *Transport Policy*, 67:40–48, 2018.
- [4] Andor Háznagy, István Fi, András London, and Tamás Németh. Complex network analysis of public transportation networks: A comprehensive study. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 371–378. IEEE, 2015.
- [5] Edward Helderop and Tony H Grubestic. Streets, storm surge, and the frailty of urban transport systems: A grid-based approach for identifying informal street network connections to facilitate mobility. *Transportation Research Part D: Transport and Environment*, 2019.
- [6] Rainer Kujala, Christoffer Weckström, Richard K Darst, Miloš N Mladenović, and Jari Saramäki. A collection of public transport network data sets for 25 cities. *Scientific data*, 5:180089, 2018.
- [7] Arnoud Mouwen. Drivers of customer satisfaction with public transport services. *Transportation Research Part A: Policy and Practice*, 78:1–20, 2015.
- [8] Pavel V Paulau, Christoph Feenders, and Bernd Blasius. Motif analysis in directed ordered networks and applications to food webs. *Scientific reports*, 5:11926, 2015.
- [9] Jari Saramäki, Jukka-Pekka Onnela, Janos Kertész, and Kimmo Kaski. Characterizing motifs in weighted complex networks. In *AIP Conference Proceedings*, volume 776, pages 108–117. AIP, 2005.
- [10] Jinjun Tang, Shen Zhang, Wenhui Zhang, Fang Liu, Weibin Zhang, and Yin Hai Wang. Statistical properties of urban mobility from location-based travel networks. *Physica A: Statistical Mechanics and its Applications*, 461:694–707, 2016.
- [11] Zhao Tian, Limin Jia, Honghui Dong, Fei Su, and Zundong Zhang. Analysis of urban road traffic network based on complex network. *Procedia engineering*, 137:537–546, 2016.
- [12] Alexandru Topirceanu, Alexandru Iovanovici, Mihai Udrescu, and Mircea Vladutiu. Social cities: Quality assessment of road infrastructures using a network motif approach. In *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 803–808. IEEE, 2014.
- [13] Sebastian Wernicke and Florian Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

ACKNOWLEDGMENTS

This work was supported by research grant GNaC2018 - ARUT, no. 1349/01.02.2019, financed by Politehnica University of Timisoara.

Versioned database storage using unitemporal relational database

Ján Janech

Faculty of Management Science and Informatics

University of Žilina
Žilina, Slovakia

jan.janech@fri.uniza.sk

Marek Tavač

Faculty of Management Science and Informatics

University of Žilina
Žilina, Slovakia

marek.tavac@fri.uniza.sk

Michal Kvet

Faculty of Management Science and Informatics

University of Žilina
Žilina, Slovakia

michal.kvet@fri.uniza.sk

Abstract—Current client-server architectures are mostly based on data sharing. It means, that individual clients must have direct access to the application server, which is, however, many times, unreachable due to connection failures. Another significant problem covers performance issues because of network latency. In this case, data are managed asynchronously and merged on demand. As a result of the deep analysis, we have proposed a new architectural design pattern of distributed information system with asynchronous data actualization. During the development, we came to the conclusion of the necessity to store versioned data on a server. This paper proposes new techniques for storing versioned data in unitemporal relational database storage. At the same time, our solution is able to preserve the advantages of RDBMS, such as referential integrity and transaction processing.

Keywords— data tuple version; temporal database; validity; references

I. INTRODUCTION

Nowadays, the strong demand for the effectivity of data approach can be recognized. Data amount and the complexity of the whole system is still rising. Data structures evolve over the time resulting in the necessity to aggregate and compose the data reflecting changing architectures. The intelligence of current information systems is largely delimited by the ability to cover and manage not only current valid states, but the whole evolution, as well. Therefore, the relational paradigm based on storing only current valid states must be replaced to handle temporal states delimited either by the physical or logical time definition. Thus, each object state is denoted by the temporal time frame expressing validity, reliability or corrections.

Temporal management originates from the relational paradigm defined in 60ties of the 20th century. With the advent of the first systems, a natural requirement to store the whole image of the data was identified. In the first phases, it was impossible to handle them, mostly due to weak technical background. First temporal aspects were defined in 70ties of the 20th century. The principle was based on object granularity, by which each data object definition was extended by the time frames, mostly expressing the validity of the particular state. Thanks to that, the primary key of the object was defined not just by the identifier itself, but also time position was necessary to be taken care of. Temporal paradigm as a core part for the validity bordering must ensure transparency and reliability, as well. Thus, each data object state cannot be represented by more than one valid state at

any time point or interval. Later, also other temporal approaches based on the granularity reflection can be identified, from the attribute perspective up to data synchronizations and object grouping. Specific group covers spatio-temporal models. [6] references real-time big data object used in smart cities of intelligent transport system. Temporal variation is covered in [7]. Temporal management using sophisticated solution to detect and optimize data layer by using object value groups is covered in [16] originating from [17] describing time-oriented applications.

Our experimental and technological environment is associated with the transport systems defining the complex environment (railway infrastructure) and individual properties of the stations, tracks, locos forming the complex operating and controlling system. Our proposed solution originates from [8] [14] [18]. As evident, individual characteristics evolve over the time and it is necessary to cover the whole complexity, to be possible to manage any time spectrum – history, current state and future, as well, whereas the data and train diagram is necessary to be operated not only in a current manner, but the changes must be reflected and approved by the historical evolution with emphasis on the efficiency. Future prognoses and plans must be handled, as well [1].

The aim of this paper is to propose solutions for the problems associated with the temporal complexity. It defines new approach joining temporal database management and versioning systems. Thus, it is not temporal database itself, nor version manager. The solution is the complex versioning system, which originates in relational database. Thanks to that, all advantages of transactions and relational paradigm is covered. Versioning is done by two levels – header and data.

Section 2 deals with the current state of the art with an emphasis on versioning technology. Data management can be divided into different structures and operated by the hierarchy. Section 3 discusses the problems and limitations of the versioning and hierarchy in the temporal environment with regards to the covering. Section 4 proposes our own techniques to solve the limitations and restrict technological problems. In section 5, impacts, properties, and implementation characteristics are defined with an emphasis on the real transport environment covered by the ZONA project (system of train management in Slovak Republic). [13] defines data management and distribution in ITS. [14] and [18] proposes technical background of the IS Zona.

II. STATE OF THE ART

Temporal modeling is a specific approach, where individual changes are monitored over time and such values

are then associated with the particular data image in unitemporal granularity [9] implemented on relational database layer [10]. A similar approach can be identified by using versions, which can be delimited either by the time or just by the order number. Temporality from that point of view is covered by the version identification and management. In principle, versioning and temporal validity can be present in the system together forming a bi-temporal approach. An individual state can be then modified by various versions [9]. The aim is to get a reliable solution [5] and optimized performance [15] forming complex ITS system [12].

When dealing with the temporal validity frames or versioning, such attributes are commonly part of the object identifier – primary key, whereas each object is evolving and particular changes are monitored and stored. To do so, the data model is extended. Individual tables are commonly represented by the composite primary key definition, which, however, provides significant problems with the integrity characterized mostly by the references. In the relational paradigm, it is said, that each foreign key value of the object can be either undefined (*NULL*), if the optional membership is defined [3] or can hold the value of the primary key, respectively its candidate expressed by the unique index, in the referenced table [2]. Fig. 1 shows the object level temporal architecture. In this case, references cannot be done directly, due to the covering problem [17]. Each foreign key object value must be covered by the primary key definition, however, it must handle time, as well. The strict rule of the referential integrity says, that also time definition must be covered. In such case, therefore, one object must be stored multiple times, if it is part of several versions consequencing in duplicities [2]. Current solutions therefore often ignore such rule and individual validity definition in the references are ignored [5]. In section 3, we cover such limitation and propose own solution based on two levels – data header and temporal definition. The first model in fig. 1 expresses conventional model with no reflection to the time. The other models are uni-temporal, time spectrum is bordered by the *BD* – begin point of the validity and *ED* – end point of the validity. As evident, *ED* does not need to be modelled, whereas each new state automatically delimits the previous image.

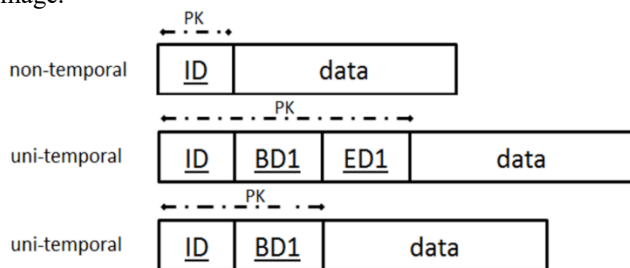


Fig. 1. Unitemporal data model [9]

Descriptive data about the environment, positions, and properties are many times defined as the complex aggregation of several attributes located even in many tables forming hierarchy. Another example is delimited by the generalization, where common parts are separated to the different structures to ensure data normalization [4] and fragment distribution [5]. In existing systems, the hierarchy can be modeled just as the cardinality 1:1 or 1:N. In the commercial environment, it is, however, often necessary to

assign a particular object to many sub-elements reflecting the validity time frame or versioning. In such case, hierarchy definition is shifted to the M:N cardinality with associative entity expressing source allocation. To distinguish version and particular association, several solutions can be used [16], unfortunately, they do not cover the time elements. Thus, referential integrity is expressed just as object correlation with no reflection to the object state versioning.

III. INTRODUCTION TO THE PROBLEMATICS

There are several problems that should be resolved before unitemporal database can be served as a store for versioned data.

The first problem reflected by the time representation in our system. Most often, datetime data types are used in standard unitemporal database approaches representing temporal validity information [10]. It is possible to use such a definition; however, it can cause problems in the following cases:

- When a system needs to identify and aggregate changes introduced together in one business transaction, it is more convenient to use other types, whereas datetime types are not powerful to represent a unique identifier.

Unique transaction identifier in unitemporal database might be required by users. For example, an administrator can want to know which changes were applied together.

- Datetime types are not suitable in the situation when additional information about the transaction is needed to be kept. The data about committing user, data origin, transaction signature, etc., can be required by the user.

As a result of those limitations and problems, another structural type is needed instead of datetime types. The only restriction is to use the ordinal type to be able to mark records with a correct validity period. This need was originally covered by the SQL/Transactional standard [11] but it was left out of the final version of the standard for some reason [3].

The second important problem is related to data integrity, which is one of the biggest advantages of relational database management systems. It is one of the reasons why to select RDBMS for data persistency management [2]. It is not necessary to implement it by the software itself. As a result, the solution saves time, money and provides better delivery and safer application.

These integrity rules are especially important for us in the RDMS:

- Entity integrity (primary keys) – as mentioned earlier, the data model must be extended. Primary keys have to include version information. It is necessary to find a solution, which will enable to uniquely identify any object time version and, at the same time, to be able to reference it from dependent objects.
- Referential integrity (foreign keys) – each object can be stored multiple times resulting in referencing problems. Which version should be referenced? There is a strict need to ensure data integrity for the whole history – all object versions.

- Uniqueness property – the aim is to ensure uniqueness property and definition across all system objects. Therefore, it is inevitable to take into account objects time versions., as well. In our solution, we do not require uniqueness across all stored property values (default behavior), but uniqueness across all stored property values for every time version is demanded separately.

Another important RDBMS advantage is protected by the support for parallel data access by multiple users with the help of the ACID transactions supported by almost all databases. The problem is most prevalent in a case of concurrent write access to the same object (mostly solved with optimistic or pessimistic locking).

In a case of versioned databases there are some limitations:

- Order of transactions is important to the users, as they represent the history of changes. Overlapped transactions cause problems we have to deal with and to be put to the correct order.
- As mentioned above, the data consistency ensured by integrity rules is different for temporal database and has to be applied for every time version separately. This is another reason to serialize transactions execution.

IV. SOLUTION PROPOSAL

As stated in the previous section, there are multiple problems with using unitemporal relational database as a versioned data storage. They can be divided into three categories: data validity identification, data integrity, and parallel access to a database.

A. Data validity period identification

As stated before, using datetime type for specifying object validity period is not enough in a versioned database. The best way is to use autoincrement integral number [11]. Each distinct value represents one version of the whole database.

To make assigning version numbers easier, “revision” table can be introduced. It uses a sequential integer as a primary key. Every time a new data is to be written to a database, a new revision is created sooner. Validity period in such a system is defined by the tuple of foreign keys referring to the revision table: `revision_from` and `revision_to`. The principle is shown in Fig. 2.

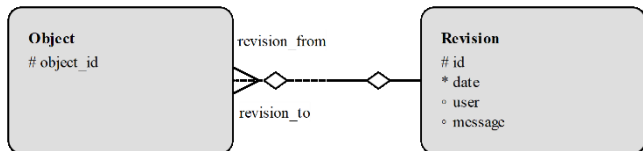


Fig. 2. Revision table

The remaining problem is how to mark the record with infinite validity. The need for this is very common in a versioned database, as the infinite validity marks currently valid data.

The simplest solution is to insert one arbitrary row to the revision table with ID holding the biggest possible number. The example of such row is in Fig. 3. The ID used here is 2^{63} , as it is the maximum possible value for the long type in the most programming languages.

id	date	user	message
9223372036854775807	9999-12-31	NULL	Max revision

Fig. 3. The maximum revision row from Revision table

No matter, which ID definition and representation is used, it enables to check validity by using BETWEEN operator in SQL, for example:

```
SELECT * FROM object
WHERE :requested_revision
BETWEEN revision_from AND revision_to;
```

B. Data integrity

There is one big advantage in the version databases in comparison to the general unitemporal databases. It should be restricted to change historical data. Changes are allowed just on the latest (head) data version.

If head data version (V1) is consistent and we will apply some consistent changes on it, we get a new head version (V2). V1 is denoted as the historical version. Because it is prohibited to change it, V1 stays consistent (as all other historical versions). So, the problem with data integrity can be reduced to ensure data integrity just on head data version and new changes applied to it.

Our proposed solution is based on the separation of history (temporal data) and current, last valid, data (head data). It is enough to apply integrity constraints (entity, referential, uniqueness) just to head data. Because one and only purpose of the head data is to ensure data consistency, we can remove all value attributes which are not used inside integrity constraints. On the other hand, temporal data contain not just the history, but also the actual version for all the attributes to simplify the model and data access. An example of transformation to this H/T table layout (head data / temporal data) can be found in Fig. 4 and Fig. 5.

The first picture contains an example of all three mentioned integrity constraint types inside the traditional non-temporal model (uniqueness is part of the primary key constraint).

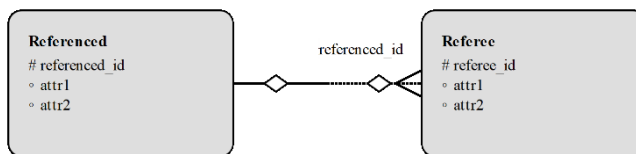


Fig. 4. Example logical data model

Based on our proposal, every table is transformed into two tables:

- H table – consists of just integrity attributes – a subset of attributes from the original model taking part in attributes used in primary, foreign keys and uniqueness constraints. The same way as the data consistency is ensured in the non-temporal model, it will be ensured for our version systems until all new data changes will be applied to these tables.

This table contains just the last data version.

- T table – consist of all original attributes to be able to store the whole history. It lacks all integrity constraints, except the new arbitrary primary key.

This table contains all temporal data.

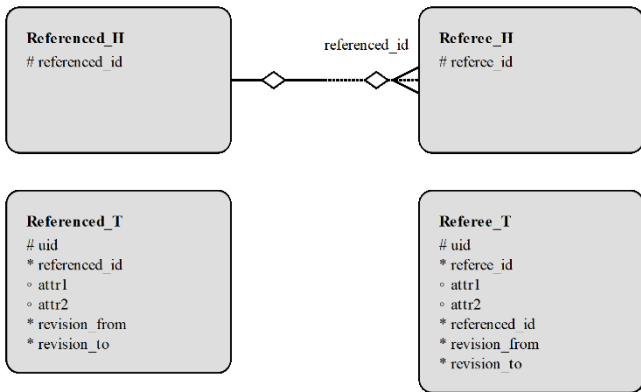


Fig. 5. Transformation to H/T table layout in the physical data model

C. Parallel access to a database.

The last serious problem we have to solve is parallel access to the database. As mentioned above, the possible solution is to serialize transaction execution.

Whereas a new record to the table Revision has to be created in advance to every transaction start (to achieve revision identifier), it is sufficient to lock revision table at transaction start and unlock it when the transaction is finished to prevent multiple transaction execution at parallel.

V. A CASE STUDY

We have verified the proposed solution by implementing it in information system ZONA (see Fig. 6). It is used by Slovak train infrastructure manager – ŽSR to plan and project train timetables in advance for the next year. It is also used to modify and store current train timetables and provide it to other systems.

Before the proposed solution was introduced into the ZONA system, much simpler solution was used. Data was saved into files using Microsoft proprietary binary serialization. To preserve the data history, the Subversion system was used. Such solution missed important advantages of RDBMS as checking data integrity, simple access to the data, possibility to manually change the data, etc. The Subversion system was not able to provide comprehensive history information as it didn't understand the binary data format.

It is a client-server system based on version database storage built on Oracle. The system is used to construct train timetables by the Slovak national infrastructure manager. The database actually contains data versions from 2011. The data model consists of 93 tables containing more than 80000 data versions (number of revisions). The biggest tables contain tens of million records (e.g. there are 18 000 000 track points with 30 000 000 versions). The whole database size is 30 GB. The example of a train history as shown in ZONA system can be seen on Fig. 7. The history was created based on the train revisions stored in versioned relational database.

Despite a long time the application is used until now and despite the amount of data stored in the database, the data have no inconsistencies. To conclude it, it can be said, that the proposed solution is robust, transparent and complexly joins the advantages of RDBMS and versioned storage systems.

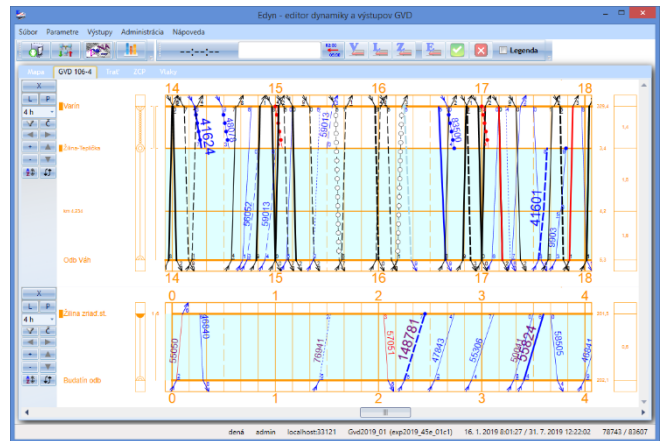


Fig. 6. The ZONA system

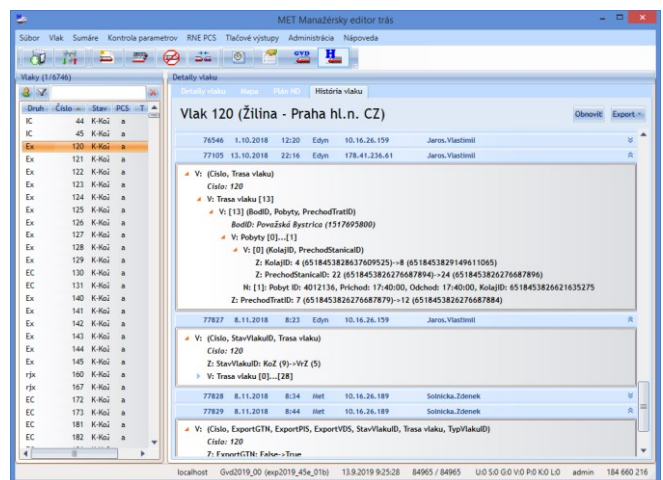


Fig. 7. An example of train version history

VI. CONCLUSION

Conventional paradigm is based on storing just current valid data. Thus, each data change automatically replaces existing data by the new version. The temporal extension is based on storing the whole evolution of individual states placed in the time sphere reflection. When data corrections are present, the problem is much deeper.

Existing solutions use the data validity definition reflected by the datetime attributes. In the real environment, however, the problem is the transparency, covering and reliability can be present. In this paper, we mostly deal with the temporal transaction definition, architecture and the whole integrity. We propose our own solution based on the header data with reflection to the primary key, referential integrity and other constraints. The descriptive data about the states are placed in the temporal table forming individual states. Thanks to that, no need to monitor time attributes is necessary to ensure.

As we stated before, the proposed solution enables to store versioned data while still retaining all advantages of RDBMS. The system ensures data consistency for all revisions, and enables access to all data revisions using standard SQL commands and tools. On top of that the system clears the strict line between the concepts of transaction and revision.

The proposed solution is experimentally tested is used in the real environment of railway transport.

ACKNOWLEDGMENT

This publication is the result of the project implementation:

Centre of excellence for systems and services of intelligent transport II., ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

The work is also supported by the *Grant system* of the University of Zilina.



"PODPORUJEME VÝSKUMNÉ AKTIVITY NA SLOVENSKU
PROJEKT JE SPOLUFINANCOVANÝ ZO ZDROJOV EÚ"

REFERENCES

- [1] M. Adolphson, "Impact on urban form by the localization of railway stations: Evidence from Sweden", *Cities*, Volume 95, 2019.
- [2] L. Ashdown, T. Kyte, "Oracle database concepts", Oracle Press. 2015
- [3] H. Darwen, "Valid time and transaction time proposals: Language design aspects," in *Temporal Databases: Research and Practice*. 1998. ISBN 978-3-540-64519-1.
- [4] B. Fekih, F. Jemili, "Distributed Architecture of an Intrusion Detection System Based on Cloud Computing and Big Data Techniques", 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunication, SETIT 2018.
- [5] J. Fernandez, J. Romera, "Indexes to Find the Optimal Number of Clusters in a Hierarchical Clustering". In 14th International Conference on Soft Computing Models in Industrial and Environmental Applications, SOCO 2019; Seville; Spain, 2020
- [6] A. M. Fernandez, et al., "Real-Time Big Data Analytics in Smart Cities from LoRa-Based IoT Networks", 14th International Conference on Soft Computing Models in Industrial and Environmental Applications, SOCO 2019
- [7] S. Hendawitharana, M. Priyasad, R. Rajapakse, "Comparative Study of Spatial and Temporal Variation of Drought Using Remotely Sensed Data". I, *Lecture Notes in Civil Engineering Volume 44*, 2020
- [8] P. Hrkút, J. Janech, E. Kršák, and M. Meško, "A New Architectural Design Pattern of Distributed Information Systems with Asynchronous Data Actualization," in *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*, pp.80–90.
- [9] T. Johnston, "Bi-temporal data – Theory and Practice", Morgan Kaufmann, 2014.
- [10] T. Johnston, R. Weis, "Managing Time in Relational Databases: How to Design, Update and Query Temporal Data". 2010. ISBN 978-0-12-375041-9.
- [11] J. Melton, "(ISO Working Draft) Temporal (SQL/Temporal)". 1999.
- [12] R. Kmet, Z. Dvorak, M. Kvet, "Map of traffic accidents", 13th International Scientific Conference on Sustainable, Modern and Safe Transport, TRANSCOM 2019
- [13] E. Krsak, "Data readers". *Communications - Scientific Letters of the University of Zilina*, Volume 18, Issue 1, 2016.
- [14] E. Krsak, P. Hrkut, P. Vestenicky, "Technical infrastructure for monitoring the transportation of oversized and dangerous goods", *Federated Conference on Computer Science and Information Systems, FedCSIS 2012*
- [15] M. Kvet, "Advanced radial approach to resource location problems", *Studies in Computational Intelligence*, Volume 718, 2018
- [16] M. Kvet, K. Matiaško, "Temporal Data Group Management", IEEE conference IDT, 2017
- [17] R. T. Snodgrass, "Developing Time-Oriented Database Applications in SQL". 1999. ISBN 1-55860-436-7.
- [18] K. Šotek, E. Kršák, H. Bachratý, V. Tavač: "New Trends of ZONA IS in environment of Slovak railways", 2011. ISBN 978-80-263-0003-8.

Detection of non-standard situation in smart water metering

Ondrej Kainz

*Department of Computers and
Informatics*

FEEI, Technical University of Košice
Košice, Slovakia

<https://orcid.org/0000-0003-0912-6710>

Eduard Karpíel

*Department of Computers and
Informatics*

FEEI, Technical University of Košice
Košice, Slovakia

eduard.karpíel@student.tuke.sk

Rastislav Petija

*Department of Computers and
Informatics*

FEEI, Technical University of Košice
Košice, Slovakia

rastislav.petija@tuke.sk

Miroslav Michalko

*Department of Computers and
Informatics*

FEEI, Technical University of Košice
Košice, Slovakia

miroslav.michalko@tuke.sk

František Jakab

*Department of Computers and
Informatics*

FEEI, Technical University of Košice
Košice, Slovakia

frantisek.jakab@tuke.sk

Abstract— In this paper the algorithm for detection of non-standard situation in smart water metering based on machine learning is designed. The main categories for non-standard situation or anomaly detection and two common methods for anomaly detection are analyzed. Proposed solution needs to fit the requirements for correct, efficient and real-time detection of non-standard situation in actual water consumption with minimal required consumer intervention to its operation. The final implemented and tested solution evaluates anomalies in water consumption for a given time in specific day and month using machine learning with semi-supervised approach.

Keywords— *anomaly detection, machine learning, non-standard situation detection, smart water metering, water consumption*

I. INTRODUCTION

Interest in smart homes has been on the rise over the recent years. One of the possible reasons is the high availability of offered solutions at affordable prices, while providing simple installation. These devices offer control of common household appliances or even lighting. They can also provide the user with various up-to-date information, such as indoor temperature or air humidity, measured by sensors incorporated in such devices. The data can be sent either to the home server outside the external network or just stored on the network, so-called “cloud” to which the user can access from any location. They can access them either through a web interface, a mobile application, or any other solution that is supported by the device.

Consumer may be interested in energy consumption. Most products are already manufactured to save energies, example of such is the water. Many households have an analogue water meter to measure the water consumption. These analog water meters contain dials on which the total water consumption is presented. The consumer alone would have to record these values and then compare the new ones to evaluate their current water consumption, e.g. per hour.

There are algorithms for image processing that can convert these dials into digital form. In addition to digitizing these values, there are also approaches for other, so-called heuristic processing. Through the usage of modern machine learning techniques, the non-standard situations may be automatically evaluated.

According to Penttilä [1], the detection of non-standard situation, or anomaly detection, represents the detection of anomalous data in a known data set. This term is closely related to both statistics and machine learning. Anomalies can be detected in various areas, such as banking (fraud detection in research by Dorj and Altangerel [2]), medicine (detection of non-standard values in health records, e.g. method presented by Carvalho et al. [3]) or information technology to detect the potential critical system failures (Hodge and Austin [4]).

Machine learning techniques depend on the context in which they are to be applied, also on the training data set, and as well whether the all types of non-standard situation are known. Detecting non-standard situation in water consumption by applying machine learning should be accurate enough to be considered reliable and correct solution with minimal human intervention.

Solution described in this paper is the component of overall solution for metering of water consumption that can be easily enabled in small and medium households that utilize analogue water meters. In the latter parts we focus only on the detection of non-standard situation from such measurements through own designed algorithm.

II. CATEGORIES FOR NON-STANDARD SITUATION DETECTION

Based on researches in [5] and [6] there are three main categories for anomaly detection: supervised, semi-supervised and unsupervised.

A. Supervised

Supervised detection assumes that a training dataset is available in which both, the normal and anomalous data is labeled. In this case, this is done by creating a prediction model for this data, which then compares them with other data and tries to determine whether they are anomalous or not. The possible issue in this case, as the authors mention in research [5], is a situation with a larger set of data, where some of them will be unknown for a given training set. Then the model will not be able to distinguish whether a given data instance is or is not anomalous. Thus, the supervised detection is mainly utilized when all types of anomalies are known, and their occurrence is uncommon.

B. Semi-supervised

The semi-supervised detection has, according to research in [6], an available set of training data with defined labels for normal data only. This, in contrast to the previous example, brings a wider range of applications, since the created prediction model can distinguish whether only a given data instance is normal or not, based on the training data. If it is not, then it is anomalous. In this case, the set of possible anomalies can be unlimited.

C. Unsupervised

Unsupervised detection does not use a training data set and is therefore most commonly used as reported in [5]. It assumes that anomalies occur less frequently than normal data in a given dataset. If this assumption is not true, techniques based on this detection will contain many false anomalies.

III. CLASSIFICATION OF METHODS FOR NON-STANDARD SITUATION DETECTION

According to research in [6] there are several methods for detection of non-standard situation. In this part two of them will be analyzed: methods of anomaly detection based on clustering and statistical methods of anomaly detection.

A. Clustering-based anomaly detection

In book [7], clustering (see Fig. 1 as an example) is described as a grouping of observation into subgroups, where these groups are much more closely interrelated than if they were linked to observations from other groups. Consequently, it is possible to look for differences between the observations.

As is described in [6] there are three different situations for anomaly detection for this method:

1. Anomalous data does not belong to any cluster.
2. Anomalous data is far from their closest centroid cluster.
3. Anomalous data belongs to small or sparse cluster.

B. Statistical anomaly detection

For statistical anomaly detection methods, according to [6], anomalies are detected based on a statistical model where the data instance is or is not matched to the model. Data instances that are less likely to be generated by the statistical model are considered anomalous. Hodge and Austin [4] categorize statistical methods into four groups: proximity-based methods, parametric, non-parametric, and semi-parametric methods.

1) Proximity-based methods

These methods are simple to implement and have no assumptions of the data distribution model. On the other hand, the time consumption exponentially increases with respect to the amount of the data, since the distance between all data is calculated.

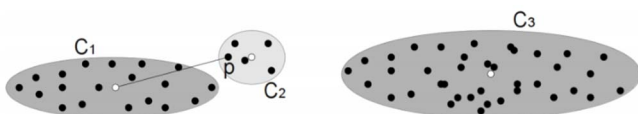


Fig. 1 Example of graphical representation of clusters. C_1 and C_3 are large clusters, C_2 is the small cluster. The distance to the centroid (marked with a white point) of the C_1 cluster is p . Image origin: [8].

The example of these methods is the k-NN algorithm, based on the closest neighbor that calculates the point distance from other points. If this distance is significantly larger, the point may be considered as anomalous. Euclidean or Mahalanobis distance may be used to calculate the distance.

As mentioned, with the increasing number of data, the time consuming of this algorithm increases exponentially. A possible solution is its optimization, which is described in detail in the research by Hodge and Austin [4].

2) Parametric methods

Parametric methods allow the statistical model to be applied very quickly to new data instances and hence are also suitable for large datasets, as described in [4]. Subsequently, they draw attention to a possible disadvantage by using a pre-selected distribution model that matches the data. However, not all data must fit into one particular model, leading to incorrect anomaly reports.

Chandola et al. in [6] present several examples of parametric anomaly detection methods, which are divided into methods based on: Gaussian model, regression model and parametric distribution mixture. Detailed analysis of techniques for these methods can be found in [4] and [6].

3) Non-parametric methods

Non-parametric methods are, according to [4], suitable for data that do not fit only in one model, respectively they can be distributed randomly and thus it is not possible to have a pre-created model for them. Thus, non-parametric methods for detecting anomalies can be used for pre-collected data and subsequently processed to determine parameters of the data where the distribution model is already known.

Two types of techniques are mentioned for non-parametric methods in [6], and are based on: histogram and kernel function. Histogram-based techniques represent the simplest technique of non-parametric methods, suitable for intrusion detection or fraud detection. For kernel function-based techniques, these functions can be used to estimate the probability distribution function for normal data instances, and new instances are considered anomalous if they are in the low probability of this function.

4) Semi-parametric methods

Semi-parametric methods, according to [4], use several local distribution models instead of one global model. It is a combination of parametric and non-parametric methods, using the already mentioned core-based methods that calculate the density. Data that lie in the low-density region is considered anomalous. The more detailed characteristics of the algorithms for these methods are described in [4].

IV. NON-STANDARD SITUATION DETECTION: THE REQUIREMENTS

It is essential to realize what requirements have to meet the proposed solution:

1. The algorithm should be able to learn the daily behavior of the user with respect to water consumption. It should be able to differentiate consumption in terms of minute, hour, day of week, and month of year. It is possible that the user has a higher daily consumption over the weekend than during the week. Also, in summer, the water

consumption may be significantly higher than at other times. Consideration should also be given to whether it is hot or cold water.

2. The algorithm should not require the user's intervention during its operation but should be able to let the user to evaluate the non-standard situation evaluated as standard. These adjustments should also affect the algorithm itself for future evaluation.
3. The algorithm should offer sufficient accurate results with a small ratio of false non-standard, respectively false standard situations.
4. The user should be able to delete all their measured data and start learning again. This may be useful if the user changes and the same device is used again.

Meeting all above mentioned requirements should be enough to implement the solution which provides accurate and effective real-time detection of non-standard situation in the current water consumption with the minimum required human intervention, yet will also provide the feedback to intervene in decision process – whether the tested value is or is not anomalous.

V. NON-STANDARD SITUATION DETECTION: THE SOLUTION

The implemented experimental solution consists of three main parts: selecting an approach for anomaly detection, design of the algorithm for anomaly detection in water consumption and the implementation of the designed algorithm.

A. Selecting an approach for anomaly detection

Based on the analysis of non-standard situation and methods for their evaluation, described in previous chapters, we distinguish three basic types of machine learning: supervised, unsupervised, and semi-supervised. All three types satisfy both the first and the fourth requirement, since they support continuous learning and the set of training data can still be erased. Supervised learning methods are highly accurate, satisfying the third requirement, but require a large set of training data, where each record has to be marked. To do this, a great effort is needed to correctly label data, which is contrary to the second requirement.

On the other hand, methods based on unsupervised learning meet this second requirement. Examples include clustering methods described in Chapter III. However, the way clusters are formed is determined by an algorithm based on the search for cluster centers. These clustering centers are determined by the algorithm itself, given the number of points around the center. Fig. 2 represents an example of a k-means algorithm with 12 clusters for water consumption data per hour on day of month-old data. In Chapter III one of three situations for anomaly detection is mentioned, where anomalous data is in small or sparse clusters. Applying this method to a twelve-cluster graph is represented by yellow and both light green clusters that show non-standard values. The same is true for the blue cluster at the top right of the graph. These clusters are somewhat more accurate for sorting of non-standard values, however still contain normal water consumption values and thus continue to produce inaccurate detection results, which does not meet the third requirement.

Methods based on semi-supervised learning use the features of supervised and unsupervised learning. They can be applied to include a training set of only standard records,

which means that during initial learning, the training set needs to be met by values from normal household behavior. At this stage, the necessary effort from the user can be expected to manipulate these values in the case of non-

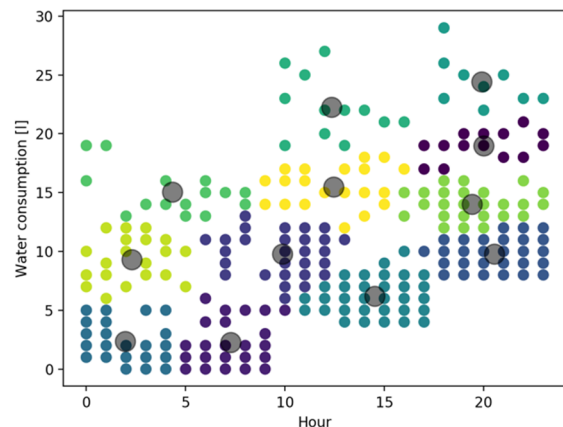


Fig. 2 Graphic representation of K-means algorithm for the current water consumption with respect to hours in day. Points are sorted into twelve clusters. Cluster centers are marked in gray.

standard situations in the learning phase. However, at the end of this phase, the algorithm will already contain a minimum set of training data to predict future water consumption values based on the same or similar properties. Then, a number of user interventions in the algorithm's learning process is minimized. That means the second requirement can be considered fulfilled. The third requirement is met, given the increasing set of training data, the accuracy of the prediction to the next measured (test) value increases.

Based on these three approaches, it is best to use the features of semi-supervised learning to meet all the requirements in this paper.

B. Design of the algorithm for anomaly detection

The first step is to get a set of training data (see Fig. 3). These training data will be stored in a remote database to which the algorithm will have direct access. Consequently, it is necessary to teach a model that will represent this semi-supervised learning. It is very important that the model learns to predict values based on only those features that are relevant to the outcome, otherwise the prediction can produce distorted results. It is also necessary to have the data at regular intervals, even in those where the consumption is zero, so that the model can correctly predict zero consumption. Then, the final consumption value can be defined by the *month*, *day*, *hour*, *minute*, and *type* features.

If the model is trained, it is ready to estimate, or to predict the next values. Only a single value representing the currently measured water consumption will be tested. The test value must be in the same form as the training data set, i.e. it must contain the same features. The resulting consumption value is not needed in this step.

If the model has predicted the future value based on the test features, the decision process follows. In this step, the predicted value is compared with the actual measured consumption. Initially, the measured value must be greater than the predicted value. Lower water consumption than expected is not seen as a negative phenomenon within households for obvious reasons. Thus, if the value is higher than the predicted value, then the relative difference must be

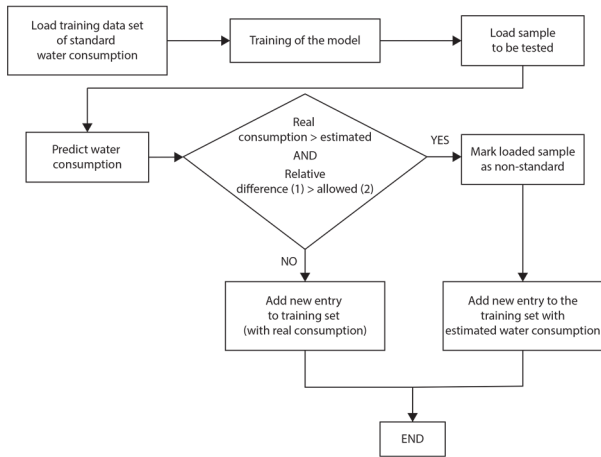


Fig. 3. Algorithm for detection of non-standard situation in water consumption

higher than the currently allowed one for the given measured value for a non-standard situation evaluation. The relative difference between the two values (a and b) is calculated as follows (see also Fig. 3 – (1) and (2)):

$$R_D = \frac{|a-b|}{\frac{a+b}{2}} \quad (1)$$

This equation was obtained from an online publication [9] by Mateusz Mucha and Álvaro Díez.

The allowed difference D , expressed in percentages, is dependent on the predicted value P and is defined as follows:

$$D = \begin{cases} (80 - 10P)\% & (80 - 10P)\% \geq 20\% \\ 20\% & (80 - 10P)\% < 20\% \end{cases} \quad (2)$$

As this equation implies, the maximum allowed difference is 80%. The allowed relative difference decreases linearly with increasing predicted value, while the minimum relative difference does not fall below 20%. This relationship improves the efficiency of detecting non-standard values, as at lower consumption values the relative difference is greater than at higher values. If the value is predicted at eight liters for a given situation, 10 liters is still acceptable, but in the case of a predicted one-liter consumption, the three-liter consumption is already too high.

The situations revealed by this equation as non-standard are stored in the database. Obviously, the training set needs to be expanded to make the other predictions more accurate. Since the goal is to leave only the standard data in this set, in this case it is not possible to save the actual water consumption. Therefore, the predicted water consumption is saved instead of the actual one. If the measured value was not marked as non-standard, it was lower than the predicted or relative difference did not exceed the allowed value, the actual water consumption is stored in the training set. Subsequently, this extended set is stored in the database.

The designed algorithm would finish in the stage of extending the training set to a new record, ready to test and evaluate other records.

C. Implementation of the designed algorithm

The algorithm is implemented in Python along with *scikit-learn* and *XGBoost* packages. Both of these packages contain

libraries that are needed to implement machine learning in a semi-supervised manner.

Three collections for MongoDB database were created:

- *measurements* – records of measured water consumption,
- *model_data* – learned data for model,
- *anomaly_data* – records of non-standard situation,
- *user_config* – user-specific configuration for learning mode.

Records that are needed to be tested against the non-standard situation are found by searching for all records in *measurements* that are not in *model_data*. The content of the entire *model_data* collection is then loaded. Since the date and time of the record is stored in a timestamp format (a non-negative natural number represented in seconds from January 1, 1970), it is necessary to extract this value to the mentioned features: *month*, *day*, *hour* and *minute*. Also, the models that allow value prediction need *DataFrame* input data that will consist of the following columns:

Month | Day | Hour | Minute | Type | Consumption

Subsequently, two frames are created from the data frame for the model - the first will contain the first four columns that will represent the features, and the second will contain the *Consumption* column, which is defined by these features. Both frames are used as input arguments for the *XGBRegressor* model from the *XGBoost* package to its saturation. After this step, the model attempts to predict actual consumption values based on new features that have been extracted from the frame for testing. The result is an array of predicted values in the decimal form, which are rounded by mathematical rules, since the actual water consumption in the system is given in whole numbers. By utilizing this field, the algorithm applies (2), where it determines whether or not a given record represents a non-standard situation. If true, an object is stored in the array of non-standard situation to represent the data structure of the measured record. A similar object is stored in another array of model data except that a rounded value of predicted consumption is used instead of current consumption.

VI. TESTING THE IMPLEMENTED SOLUTION

To validate that the experimental solution meets the required criteria it needs to be tested. The testing of the algorithm consisted of two parts. First, the actual consumption prediction was tested. Then, the detection of non-standard situation was tested.

A. Testing the prediction of actual water consumption

Prediction of values is part of the solution for the evaluation of non-standard situation. The overall output of this algorithm is tested, i.e. comparing predicted values to actual ones. Two test cases were included for model data:

1. Testing of data prediction with model data per one week.
2. Testing of data prediction with model data per one month.

Both cases were tested for the seven days awaiting evaluation. Tables I and II show the results for 22nd hour on the second test day to find significant differences. It can be

seen that the predictions for monthly learning are more accurate when compared to weekly learning. It is also supported by the fact that the average difference between the predicted value and the actual value is 0.33 liters after the monthly learning and 0.4 liters after the weekly learning on a given sample. The percentage success rate that the predicted value was the same as the actual value is 64% after weekly learning, and 70% after monthly learning. This comparison demonstrates that with increasing learning time, the accuracy of the prediction of measured values improves.

B. Testing the non-standard situation evaluation

The marking of the currently measured value as non-standard depends on the prediction of the values for the given parameters or features. As in the previous testing, both model data and weekly and monthly data was tested here as well.

Sixteen non-standard situation arose in the test data for the first week of March. Tables III and IV show a list of the first ten detected non-standard situation. These include the day and time of the non-standard situation, the type of water, the predicted and actual consumption, and whether it was or was not actual non-standard situation.

TABLE I. SAMPLE DATA OUTPUT FROM TESTING OF WATER CONSUMPTION PREDICTION VALUES WITH MODEL DATA PER WEEK.

Date	Day	Time	Water type	Predicted consumption [in l]	Real consumption [in l]
02.03.19	Sat	22:03	Cold	0	1
02.03.19	Sat	22:04	Hot	0	0
02.03.19	Sat	22:19	Cold	0	1
02.03.19	Sat	22:20	Hot	0	0
02.03.19	Sat	22:31	Cold	0	1
02.03.19	Sat	22:32	Hot	0	1
02.03.19	Sat	22:47	Cold	1	1
02.03.19	Sat	22:48	Hot	0	0

TABLE II. SAMPLE DATA OUTPUT FROM TESTING OF WATER CONSUMPTION PREDICTION VALUES WITH MODEL DATA PER MONTH.

Date	Day	Time	Water type	Predicted consumption [in l]	Real consumption [in l]
02.03.19	Sat	22:03	Cold	1	1
02.03.19	Sat	22:04	Hot	0	0
02.03.19	Sat	22:19	Cold	1	1
02.03.19	Sat	22:20	Hot	0	0
02.03.19	Sat	22:31	Cold	1	1
02.03.19	Sat	22:32	Hot	0	1
02.03.19	Sat	22:47	Cold	1	1
02.03.19	Sat	22:48	Hot	0	0

By comparing these tables, it can be seen that the evaluation of non-standard situation for model data per one month detected more false positives than model data per one week. However, the total number of non-standard situation found was 23 after the monthly learning, and 29 after one-week learning, while all the actual non-standard situation found in the weekly learning were also found in monthly

TABLE III. OUTPUT OF THE FIRST TEN OUTPUT RECORDS OF TESTING THE EVALUATION OF NON-STANDARD WATER CONSUMPTION SITUATIONS WITH MODEL DATA PER WEEK.

Date	Day	Time	Water type	Predicted consumption [in l]	Real consumption [in l]	Non-standard
01.03.19	Fri	12:20	Hot	0	2	No
01.03.19	Fri	20:16	Cold	1	3	Yes

Date	Day	Time	Water type	Predicted consumption [in l]	Real consumption [in l]	Non-standard
01.03.19	Fri	23:33	Hot	0	2	No
02.03.19	Sat	10:33	Hot	0	2	No
02.03.19	Sat	11:02	Cold	1	3	Yes
02.03.19	Sat	13:02	Hot	0	3	Yes
02.03.19	Sat	20:20	Hot	0	2	No
02.03.19	Sat	20:34	Hot	0	3	Yes
02.03.19	Sat	10:17	Cold	1	3	Yes
02.03.19	Sar	20:20	Hot	0	2	No

TABLE IV. OUTPUT OF THE FIRST TEN OUTPUT RECORDS OF TESTING THE EVALUATION OF NON-STANDARD WATER CONSUMPTION SITUATIONS WITH MODEL DATA PER MONTH.

Date	Day	Time	Water type	Predicted consumption [in l]	Real consumption [in l]	Non-standard
01.03.19	Fri	12:03	Hot	0	2	No
01.03.19	Fri	12:20	Hot	0	2	No
01.03.19	Fri	12:33	Hot	0	2	No
01.03.19	Fri	20:16	Cold	1	3	Yes
01.03.19	Fri	22:48	Hot	0	2	Yes
01.03.19	Fri	23:33	Hot	0	2	No
02.03.19	Sat	10:33	Hot	0	2	No
02.03.19	Sat	11:02	Cold	0	3	Yes
02.03.19	Sat	13:02	Hot	0	3	Yes
02.03.19	Sar	20:47	Hot	0	2	No

learning. In addition, Friday's record at 22:48 (Tab. IV) was not detected as a non-standard situation for one-week model data, however, was detected on model data for one month, which was evaluated correctly. A total of 16 actual non-standard situation detected 14 at weekly, and 15 at monthly learning. In both cases there was a false negative evaluation of the records, yet following the results of this test it can be said that longer-term learning produces more accurate results, with fewer false positives and also false negatives.

VII. CONCLUSION

In this paper, the experimental solution for detection of a non-standard situation in a current water consumption based on machine learning is described. Thanks to semi-supervised approach, the described solution offers a quite accurate prediction for future water consumption. And based on the predicted value, an evaluation is made, whether the measured value does or does not represents the non-standard situation with respect to hour in day, day in week and the current month. The evaluation process uses the percentage difference between the predicted value and the actual one. Also, the allowed percentage difference for a given predicted value is calculated. If the calculated percentage difference is greater than allowed, the test value is evaluated as non-standard.

This proposal was tested on two scenarios – with learned model data for a week and with learned model data for a month. Test results have shown that longer-term learning provides a more accurate prediction and fewer false-positive and false-negative findings of non-standard situation. Accordingly, the described solution is sufficiently effective and accurate for a real-time detection of non-standard situation in the current water consumption.

REFERENCES

- [1] J. Penttilä, "A method for anomaly detection in hyperspectral images, using deep convolutional autoencoders," 2017.
- [2] E. Dorj and E. Altangerel, "Anomaly detection approach using hiddenmarkov model," in *Ifostr*, vol. 2. IEEE, 2013, pp. 141–144.

- [3] L. Carvalho, C. Teixeira, E. C. Dias, W. Meira, and O. Carvalho, "A simple and effective method for anomaly detection in healthcare," in *Proceedings of the SIAM International Conference on Data Mining Workshop*, vol. 2015, 2015, pp. 16–24.
- [4] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.
- [5] C. Geijer and J. Andreasson, "Log-based anomaly detection for system surveillance," Ph.D. dissertation, Masters thesis, Chalmers University of Technology, Gothenburg, Sweden, 2015.
- [6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [7] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.
- [8] M. Amer and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer," in *Proc. of the 3rd Rapid-Miner Community Meeting and Conference (RCOMM 2012)*, 2012, pp. 1–12.
- [9] M. Mucha and Á. Díez. Percentage difference calculator. [Online]. Available: <https://omnicalculator.com/math/percentage-difference>

Valuation in the Sector of New Technologies – Taxonomic Measures and Comparison of Markets

Paweł Kossecki

Polish National Film, Television and Theater School

ul. Targowa 61/63

90-323 Łódź, Poland

kossecki@poczta.onet.pl

Together with the changes introduced by IT technologies to the traditional economy, a new economy appeared. The term “new economy” emerged in the 1990s. It was associated with the changes IT technologies introduced to the traditional economy. The speed of the development of the sector of the new economy on the one hand, and uncertainty on the other hand, were associated with the highly speculative attitude of stock-market investors and the fast development of Venture Capital and Private Equity funds.

Most of the telecommunication companies should be considered as companies using the so-called subscription-based business model, where the customer pays a fixed subscription fee for access to a product or service.

Common problem in valuation of telecommunication companies is finding comparable data and markets for valuation. The objective of the investigations presented in this paper is the recognition and classification of similarities and differences between different countries related to the market development. For this purpose, taxonomic measures, especially multivariate comparative statistical analysis methods were used. Author presented a method for comparison of the markets based on cluster analysis. The study covers selected European countries. Presented method can be used for comparing markets in cases of enterprise valuations sector and comparing the whole markets indicating their level of development.

Based on presented method, five subgroups in Europe can be distinguished. Their members can be considered as comparable markets. There are significant differences between the Central-Eastern European countries. They belong to two subgroups.

Keywords — *Telecommunication, valuation, taxonomy, taxonomic measures*

I. INTRODUCTION

The telecommunication market has undergone tremendous changes, mainly due to the introduction of new technologies. Once just providing traditional analogue voice transmission, now attract their customers with high-speed Internet access, mobile telephony, TV retransmission and other services delivered at very competitive prices.

This sector was closely examined by private equity investors. Currently telecomm operators work in a high competitive as well as still fragmented market. Strong competition is due to the development of traditional telecoms, satellite platforms, as well as cable operators.

Many operators developed mainly due to the acquisitions. It increased the leverage of the sector and demanded pressure on operational management and optimizing financial results by maximizing the efficiency of the sales process and capital expenditures.

Telecomm operators should be considered as companies using the so-called subscription-based business model, where the customer pays a fixed subscription fee for access to a product or service. The level of monthly ARPU (*average revenues per user*) from basic telecommunication service is rather low in most of the EU countries. Moreover, there is a large dispersion between Western and Central-Eastern Europe; specific for the Central-Eastern European countries is the quite low nominal ARPU.

Current models for creating value are strongly related to concept of CLV (*Customer Lifetime Value*) mainly due to the high customer acquisition costs [12], [13]. It creates the necessity for very careful evaluations of the profitability of the clients during their whole lifetime. The most popular indicators, which characterize the economic situation of the enterprise in the telecommunication sector, usually taken into consideration during the valuation, are ARPU and EBITDA margin (EBITDA - *earnings before interest and tax, depreciation and amortization*). Telecommunication companies must very well understand the structure of their client's revenues and costs, to maximize revenues and margins from all services. Telecoms are typical CAPEX (*capital expenditures*) driven companies, so the level of prices for service is most important factor influencing profitability and level of valuation [13], [14].

Three types of basic services are being sold:

- access to the Internet,
- telecommunication service,
- TV service.

In addition, telecommunication companies sell a whole range of supplementary services.

In cases of European operators, ARPU is under strong pressure due to the general tendency to cut prices and the willingness of clients to move to cheaper sets of products.

The basic way to increase ARPU is the sale of additional services, as well as offering new sets of products with more competitive prices. Strong, parallel competition in the broadband Internet sector results in high pressure to cut prices [19].

To speed up broadband Internet access in rural areas, some countries, like Poland, are starting to subsidize broadband initiatives; however, in most cases, cable operators have not penetrated this market. These initiatives are the main competition for digital platforms, which operate mainly outside the big cities.

In the case of operators working in small markets, a common problem is finding comparable data for valuation.

Author presented a model for comparison of the markets based on cluster analysis. This method can be used for comparing markets in cases of enterprise valuations in the telecommunication sector.

Results of the telecommunication market analysis may be very useful in the case of valuation of small telecommunication companies operating in the Central-Eastern Europe. In some countries, there is a lack of data related to comparable companies and international comparisons must be made.

With organic growth opportunities narrowing, telecom players need a better way to screen potential M&A targets [3].

When conducting a valuation based on multiples, the biggest challenge is picking the right economic parameters as the multiples, which help to identify key value drivers of the enterprise and choose proper markets to compare [14], [15].

The most popular ratios for valuation using market comparison are [12], [14]:

- P/E – Price Earnings Ratio
- P/BV – Price to Book Value Ratio
- P/S – Price-Sales Ratio
- EV/EBITDA
- EV/number of subscribers

EV/EBITDA is the most common ratio in the telecommunication sector and most widely used multiple based on enterprise value (EV). EBITDA is largely unaffected by accounting differences and is neutral to the capital structure of the company. Quite commonly in the sector is used also EV/number of subscribers. In both cases problem is to find comparable companies and comparable markets. When comparing companies operating in various markets, it is necessary to note differences in business/economy cycles, which particularly strongly influence the growth rate and future revenues.

II. METHODOLOGY OVERVIEW

The difficulties with using market comparison for financial valuations are due to the problem of collecting accurate data in different countries, and finding comparable markets. In the analysis shown below, market comparison methodology supported by taxonomic measures was applied.

It is usually possible to find markets with higher and lower levels of EBITDA as well as margin and the most important problem is to compare markets with objective criteria.

The objective of the investigations presented in what follows is the recognition and classification of similarities and differences between European countries, related to the telecommunication market. For this purpose, the formalism and methods of multivariate comparative statistical analysis will be used. These methods are necessary tools to be applied in quantitative analysis of objects represented by many variables [4].

The result of the quantitative comparative analysis is, in general, the appropriate grouping of the objects considered, that is partition into groups of similar objects, significantly different from objects assigned to other groups. The notion of similarity is connected with the notion of distance between objects. The number of groups and their characteristics are not given in advance. The objective is rather to reveal and classify the existing similarities and dissimilarities [4].

The method of *K-means* is described and will be used in the analysis presented in what follows.

Organization of Data and Clustering Method

In the frames of the multivariate statistical analysis, the set of data represents, in general, measurements of many variables related to the set of objects considered. Assume that measurement refers to m variables; $m \geq 2$. It is to be represented by the vector-function:

$$\mathbf{X} = [X_1, X_2, \dots, X_m] \quad (1)$$

Consider now measurements on the set of n objects. Measurement on the object i is to be represented by the point vector \mathbf{x}_i :

$$\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}, \quad (i = 1, 2, \dots, n) \quad (2)$$

Measurements on the set of n objects are to be represented by the $n \times m$ matrix:

$$\mathbf{X} = [x_{ij}] \quad (3)$$

where x_{ij} is the measurement of the j -th variable on the i -th object.

Measurements of different variables are expressed, in general, in different units. Most methods of the multivariate comparative analysis may be applied if measurements are given in the same units and are of comparable order. Several algorithms have been proposed in the literature for clustering [1]:

- ISODATA [3], [11],
- CLARA [11],
- CLARANS [16],

- Focusing Techniques [6],
- P-CLUSTER [10],
- DBSCAN [5], Eijcluster [8],
- BIRCH [20],
- GRIDCLUS [18].

Different normalization procedures can be applied. The most common of them is standardization of the variables [5]:

$$Z_j = \{z_{1j}, z_{2j}, \dots, z_{nj}\}, \quad (j = 1, 2, \dots, m) \quad (4)$$

$$z_{ij} = \left(\frac{x_{ij} - \bar{x}_j}{s_j} \right), \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \quad (5)$$

where:

z_{ij} – standardized value of the variable X_j on the i -th object,

\bar{x}_j – arithmetic average of the variable X_j ,

s_j – standard deviation of the variable X_j .

The mean value of the standardized variable is zero and its standard deviation is 1. Standardized data allows us to easily distinguish objects, which are below or above average with relation to specific variables.

To compare items described by many variables, the notion of similarity - and dissimilarity - is necessary and must be formally defined. In the multivariate comparative analysis, the measure of dissimilarity is the distance between objects, represented as points in the space of standardized variables. It is called statistical distance [5]. Most commonly used is the Euclidean distance between objects:

$$d_{il} = \sqrt{\sum_{j=1}^m (z_{ij} - z_{lj})^2}, \quad (i, l = 1, 2, \dots, n) \quad (6)$$

where:

d_{il} – distance between objects i and l ,

z_{ij} – normalized value of the variable X_j on the i -th object.

K-means method

The idea of the *K-means* method is the partition of a set of objects, represented by points in the space of standardized variables, into subsets (clusters) of similar elements, concentrated around the nearest centers (means). Partition into K clusters C_k is to be realized by minimization of the function G , which represents overall scattering of points within clusters:

$$\{C_k\}, (k = 1, 2, \dots, K) \rightarrow \min G$$

$$G = \sum_{k=1}^K \sum_{i \in C_k} \sum_{j=1}^m (z_{ij} - \gamma_{kj})^2 \quad (7)$$

where γ_k is the point vector representing mean position of objects assigned to the cluster C_k ; it is called center of gravity of the cluster:

$$\gamma_{kj} = \frac{1}{n_k} \sum_{i \in C_k} z_{ij} \quad (8)$$

The function G may be thus represented in the form:

$$G = \sum_{k=1}^K \sum_{i \in C_k} (g_{ik})^2 \quad (9)$$

where g_{ik} is the Euclidean distance between the object i , which belongs to the cluster C_k , and the center of gravity of this cluster.

The appropriate choice of k is problem and domain dependent and generally a user tries several values of k [1].

The final number of clusters may be specified in advance or determined as part of the clustering procedure. In this paper, the second approach will be used.

Finding a partition into clusters that corresponds to the minimum of the overall scattering function is a mathematical task – to be solved numerically - that usually has a specific solution (you can easily come up with points distributions for which there is no single solution).

III. PARTITION OF EU COUNTRIES INTO GROUPS IN VIEW OF THEIR ECONOMIC CONDITIONS AND TELECOMMUNICATION MARKETS

To perform a quantitative comparative analysis of the telecommunication market, unified and complete data for a sufficiently large set of countries are necessary.

Aggregated data related to the telecommunication market and main statistics for a group of twenty-one selected European countries are presented in tab.1. The following basic variables describing the market were considered:

- Real Gross Domestic Product *per capita* (EUR, GDP *per capita*);
- Average monthly service cost for the speed range Internet from 30 Mbit/s to 100 Mbit/s (EUR, including VAT tax);
- Monthly basket values for a moderately active user of telecommunication services (EUR).

Gross Domestic Product (GDP) per capita must be taken into account in analysis of the telecommunication market because it determines to a great extent the economic conditions in which telecommunications operate in a given country. Average monthly service cost for the speed range Internet and monthly basket values for a moderately active user of telecommunication services describe level of prices and indirectly level of profitability. All variables are in the same units and described method can be applied.

A number of possible partitions of twenty-one countries into clusters were examined, using three standardized variables. Partition into three, four and five clusters were considered. There is low correlation coefficient between the GDP *per capita* and monthly service cost for the speed range Internet (0,2556) as well as basket values of telecommunication services (0,0141).

TABLE I. BASIC PARAMETERS RELATED TO THE TELECOMMUNICATION MARKET IN SELECTED EUROPEAN COUNTRIES (2017) [7], [17]

# of case	Country	Average monthly service cost for the speed range Internet (EUR)	Real GDP (EUR)	Monthly basket values for a moderately active user of telecomm. services (EUR)
1	Latvia	12,68	11600	52,70
2	Greece	34,52	17400	40,64
3	Slovenia	35,75	19400	27,36
4	Slovakia	15,96	15000	37,24
5	Ireland	36,99	56400	46,95
6	Hungary	22,70	11800	23,60
7	Denmark	22,87	47100	24,36
8	Germany	22,90	35500	34,76
9	Poland	22,98	11800	29,94
10	Belgium	23,63	35000	20,57
11	Czechia	24,63	17200	20,76
12	Finland	26,05	35900	26,07
13	Italy	26,40	26400	40,38
14	Netherland	27,81	48700	39,44
15	Sweden	28,42	42800	27,87
16	Estonia	28,47	14600	26,63
17	Spain	46,92	24500	50,50
18	Austria	29,48	37200	51,79
19	Portugal	29,63	17500	60,91
20	France	31,13	32400	40,28
21	Great Britain	34,32	32200	43,89
	Average	27,82	28114	36,51

TABLE II. PARTITION OF A GROUP OF TWENTY-ONE EUROPEAN COUNTRIES INTO FIVE CLUSTERS [OWN SOURCE]

Country	# of case	# of cluster	Statistical Dis-tance from the Cluster Center
Latvia	1	1	0,040453
Slovakia	4	1	0,040453
Greece	2	2	0,048023
Italy	13	2	0,067953
Spain	17	2	0,158435
Portugal	19	2	0,178286
France	20	2	0,053782
Great Britain	21	2	0,028568
Denmark	7	3	0,037157
Germany	8	3	0,049703
Belgium	10	3	0,033446
Finland	12	3	0,007278
Sweden	15	3	0,018387

Slovenia	3	4	0,078318
Hungary	6	4	0,022667
Poland	9	4	0,029390
Czechia	11	4	0,021661
Estonia	16	4	0,002732
Ireland	5	5	0,066946
Netherland	14	5	0,038887
Austria	18	5	0,075585

TABLE III. PARTITION OF A GROUP OF TWENTY-ONE EUROPEAN COUNTRIES INTO FOUR CLUSTERS [OWN SOURCE]

Country	# of case	# of cluster	Statistical Dis-tance from the Cluster Center
Latvia	1	1	0,047164
Slovakia	4	1	0,114821
Portugal	19	1	0,162159
Ireland	5	2	0,165332
Netherland	14	2	0,111056
Spain	17	2	0,24692
Austria	18	2	0,046435
France	20	2	0,044886
Great Britain	21	2	0,021753
Greece	2	3	0,108107
Slovenia	3	3	0,059253
Hungary	6	3	0,060834
Poland	9	3	0,034007
Czechia	11	3	0,060625
Italy	13	3	0,11401
Estonia	16	3	0,009568
Denmark	7	4	0,037157
Germany	8	4	0,049703
Belgium	10	4	0,033446
Finland	12	4	0,007278
Sweden	15	4	0,018387

TABLE IV. PARTITION OF A GROUP OF TWENTY-ONE EUROPEAN COUNTRIES INTO THREE CLUSTERS [OWN SOURCE]

Country	# of case	# of cluster	Statistical Dis-tance from the Cluster Center
Latvia	1	1	0,047164
Slovakia	4	1	0,114821
Portugal	19	1	0,162159
Ireland	5	2	0,246326
Denmark	7	2	0,214746
Germany	8	2	0,059866
Finland	12	2	0,116671
Italy	13	2	0,082798
Netherland	14	2	0,061557
Sweden	15	2	0,086809

Country	# of case	# of cluster	Statistical Distance from the Cluster Center
Spain	17	2	0,41258
Austria	18	2	0,10541
France	20	2	0,018209
Great Britain	21	2	0,047347
Greece	2	3	0,155159
Slovenia	3	3	0,058497
Hungary	6	3	0,047495
Poland	9	3	0,042908
Belgium	10	3	0,180023
Czechia	11	3	0,032101
Estonia	16	3	0,007235

Results of the partition represented in standardized variables are presented in tab. 2, 3, 4. Distances between clusters centroids are listed in tab. 5, 6, 7. Standardized distances between members of particular clusters and cluster centroids are in all cases smaller than distances between centroids. For calculations *Statistica* software was used.

TABLE V. STANDARDIZED DISTANCES BETWEEN CLUSTER CENTROIDS – PARTITION FOR FIVE CLUSTERS [OWN SOURCE]

	C ₁	C ₂	C ₃	C ₄	C ₅
C ₁	0,00000	0,39411	0,63353	0,36567	0,83084
C ₂	0,39411	0,00000	0,40083	0,34846	0,25414
C ₃	0,63353	0,40083	0,00000	0,29879	0,30074
C ₄	0,36567	0,34846	0,29879	0,00000	0,79863
C ₅	0,83084	0,25414	0,30074	0,79863	0,00000

TABLE VI. STANDARDIZED DISTANCES BETWEEN CLUSTER CENTROIDS – PARTITION FOR FOUR CLUSTERS [OWN SOURCE]

	C ₁	C ₂	C ₃	C ₄
C ₁	0	0,49040	0,31939	0,66598
C ₂	0,49040	0	0,41828	0,29598
C ₃	0,31939	0,41828	0	0,2628
C ₄	0,66598	0,29598	0,2628	0

TABLE VII. STANDARDIZED DISTANCES BETWEEN CLUSTER CENTROIDS – PARTITION FOR THREE CLUSTERS [OWN SOURCE]

	C ₁	C ₂	C ₃
C ₁	0,00000	0,45540	0,39309
C ₂	0,45540	0,00000	0,28830
C ₃	0,39309	0,28830	0,00000

As a preferred solution, which will be discussed below, a partition into five clusters was selected.

This solution is distinguished by the fact that any reassignment of items significantly increases the overall scattering of points within clusters.

Cluster 1 constitutes two Central-Eastern European countries: Latvia and Slovakia. Both countries have quite low average monthly service cost for the speed range Internet and quite high cost of monthly basket values for a moderately active user of telecommunication services.

Cluster 4 includes rest of the region: Slovenia, Hungary, Poland, Czechia, Estonia. There is higher cost of speed Internet than in case of countries in cluster 1, but more expensive telecommunication services. In both clusters are countries with similar level of GDP *per capita*.

Cluster 2 includes Mediterranean countries: Greece, Italy, Spain, Portugal as well as France and Great Britain. There are countries with different level of GDP *per capita* and prices of Internet access and telecommunication services. In some cases (Portugal and Spain) are high prices of telecommunication services and lower GDP *per capita*.

Cluster 3 constitutes: Denmark, Germany, Belgium, Finland and Sweden. In those countries prices of telecommunication services and Internet access are not significantly higher in comparison to Central-Eastern European countries classified in cluster 4, but there is significantly higher GDP *per capita*.

Cluster 5 constitutes: Ireland, Netherland and Austria. Those countries have high prices of Internet access, telecommunication services and high GDP *per capita*.

VI CONCLUSIONS

In the group of twenty-one European countries, several subgroups can be distinguished, whose members can be considered as comparable markets to the telecommunication markets. There are significant differences between the Central-Eastern European countries. They belong to two subgroups:

- Latvia, Slovakia;
- Slovenia, Hungary, Poland, Czechia and Estonia.

This division is due to the difference in prices of speed Internet (cheaper in Latvia and Slovakia) and telecommunication services (more expensive in Latvia and Slovakia).

Western European countries constitute 3 subgroups:

- Greece, Italy, Spain, Portugal, France and Great Britain;
- Denmark, Germany, Belgium, Finland and Sweden;
- Ireland, Netherland and Austria.

There is significant difference in prices of speed Internet and telecommunication services and they are not related to the level of GDP *per capita*.

In some relatively rich countries, level of prices is comparable to Central-Eastern Europe.

Described method was used by one of author to classification of telecommunication markets for valuation purposes. In some countries, there is a lack of data related to

comparable companies and international comparison must be applied.

Method is useful mainly, because comparison is based on public data and outcome is related to economic profits.

It can help telecom companies to choose proper markets to find acquisition targets.

REFERENCES

- [1] K. Alsabti, S. Ranka, V. Singh, "An efficient k-means clustering algorithm", Syracuse University, Electrical Engineering and Computer Science. 43. <https://surface.syr.edu/eecs/43>, 1997
- [2] O. Bosch, S. Nagel, J. Tiefel, "A better way to M&A: Zeroing in on telecom value" <https://www.mckinsey.com/industries/telecommunication/our-insights/a-better-way-to-m-and-a-zeroing-in-on-telecom-value>, November 2016, Telecommunication Insights, McKinsey & Company
- [3] R. C. Dubes, A. K. Jain, "Algorithms for Clustering Data". Prentice Hall, 1988
- [4] J. Dziechciarz, "Econometrics. Methods, examples, exercises", Wrocław University of Economics Publishing House, Wrocław, 2003
- [5] M. Ester, H. Kriegel, J. Sander, X. Xu, "A DensityBased Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, August 1996
- [6] M. Ester, H. Kriegel, X. Xu, "Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification", Proceedings of the Fourth International Symposium on Large Spatial Databases, 1995
- [7] Eurostat, GDP per capita at market prices, https://ec.europa.eu/eurostat/web/products-datasets/-/sdg_08_10, 2019
- [8] J. Garcia, J. Fdez-Valdivia, F. Cortijo, R. Molina, "Dynamic Approach for Clustering Data", *Signal Processing*, 44(2), 1994
- [9] R. A. Johnson, D. W. Wichern, "Applied Multivariate Statistical Analysis". 6th ed., Prentice-Hall, Englewood Cliffs, NJ, 2007
- [10] D. Judd, P. McKinley, A. Jain, "Large-Scale Parallel Data Clustering", Proceedings of International Conference on Pattern Recognition, August 1996
- [11] L. Kaufman, P. J. Rousseeuw, "Finding Groups in Data: an Introduction to Cluster Analysis", John Wiley & Sons, 1990
- [12] P. Kossecki, „Kreowanie i pomiar wartości przedsiębiorstwa w świecie Internetu”, PWSFTViT Publishing House, Łódź, 2011
- [13] P. Kossecki, „Powiązanie zarządzania wartością przedsiębiorstwa i zarządzania sprzedażą wśród operatorów kablowych”, *Problemy Zarządzania*, 1/2012 (35), p. 164-178, 2012
- [14] P. Kossecki, "Valuation and Value Creation of Cable TV Operators. Cable Operators and Copyright Fees", PWSFTViT Publishing House, Łódź, 2015
- [15] P. Kossecki, K. Kompa, "Valuation of cable operators – decision tree and synthetic measure of development method (SMR) for market comparison", 80th International Atlantic Economic Conference, 9-11.10.2015, Boston
- [16] R.T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining", Proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, p. 144–155, 1994
- [17] Office of Electronic Communications, "Report on the state of the telecommunications market in Poland in 2017", https://www.uke.gov.pl/download/gfx/uke/en/default/akualnosc/36/16/1/raport_o_stanie_rynku_telekomunikacyjnego_en.pdf
- [18] E. Schikuta, "Grid Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets", Proceedings of the 13th International Conference on Pattern Recognition, 2, 1996
- [19] Solon, "Solon Study Broadband on Demand", 2011
- [20] T. Zhang, R. Ramakrishnan, M. Livny. "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, p. 103–114, June 1996

Short-Term Rainfall Estimation by Machine Learning Methods

Peter Krammer

*Department of Parallel and Distributed
Information Processing
Institute of Informatics,
Slovak Academy of Sciences
Bratislava, Slovakia
peter.krammer@savba.sk*

Marcel Kvassay

*Department of Parallel and Distributed
Information Processing
Institute of Informatics,
Slovak Academy of Sciences
Bratislava, Slovakia
marcel.kvassay@savba.sk*

Ondrej Habala

*Department of Parallel and Distributed
Information Processing
Institute of Informatics,
Slovak Academy of Sciences
Bratislava, Slovakia
ondrej.habala@savba.sk*

Ladislav Hluchý

*Department of Parallel and Distributed
Information Processing
Institute of Informatics,
Slovak Academy of Sciences
Bratislava, Slovakia
ladislav.hluchy@savba.sk*

Abstract — Modeling stochastic systems is a real challenge in many areas. Even the meteorology domain is not an exception; the modeling of precipitation activity is markedly stochastic and is influenced by a number of related physical variables (temperature, pressure, humidity, wind). Accurate precipitation estimation is thus highly non-trivial. Today's technical capability, automated data measuring (whether using Radar or automatic meteorological stations), as well as subsequent large-scale data processing and regression model training, allow the meteorological estimations and predictions with increasing accuracy. This paper demonstrates selected uses of artificial neural networks in the field of meteorology, as well as solving problems with pre-processing and integrating time-spatial meteorological data.

Keywords — *Neural network, Machine learning, Data mining, Modeling, Meteorology, Rainfall estimation, Data integration, Data cleaning, Time-spatial data*

I. INTRODUCTION

The application of machine learning methods and artificial intelligence has recently intensified in many domains like economics, medicine, astronomy, hydrology and others. One such area is meteorology, although the modelling of its strongly stochastic processes is highly non-trivial. Nevertheless, such models are in high demand for the prediction of weather-related risks in agriculture, for large-scale cultural or sport-related public events, etc. Accurate modelling in meteorology is thus an ongoing challenge.

II. OVERVIEW

There are many publications that tried to model precipitation. But most of them focused on locales with frequent or heavy rains, such as India, Bangladesh and other (sub)tropic countries affected by monsoon rains (see Fig.1) or, alternatively, on coastal areas.

The reason is that meteorological conditions in these areas are vital for their economy as they strongly affect agriculture, tourism, power industry, etc. and may bring about various large-scale threats, such as floods, tornadoes, etc.

Several publications on hydro-meteorology, such as [14, 15], focused on the role of rainfall estimation using meteorological radar, thus modeling the volume of

precipitation based on radar reflectivity. They used specific physical and meteorological approaches, often designed just for that purpose.

The publication [13] presented models of raindrop distribution and analyzed the magnitude of radar reflectivity and its errors, as well as many other aspects. These analytical steps are indispensable when using strictly physical / meteorological models, but they have also proved to be very helpful in our machine learning approach. Specifically, we have used them to estimate the vertical speed of raindrops (taking into account the influence of the wind), as well as the conversion of the radar reflectance quantity into water volume in the atmosphere.

Conversely, however, several IT solutions and approaches were primarily focused on the task of predicting the future rainfall, which was of a slightly different nature and used different sophisticated approaches, very often relying on neural networks. Paper [12] provided a solid and wide overview of rainfall prediction; it described more than 30 researches typically tending towards machine learning rather than physics-based approaches.

The vast majority of articles tried to predict rainfall over large periods of time (weeks, months, or even years). An interesting novel hybrid approach was defined in 2019 [9], which combined feedforward artificial neural network with seasonal decomposition and discrete wavelet transform for monthly rainfall prediction in Vietnam.

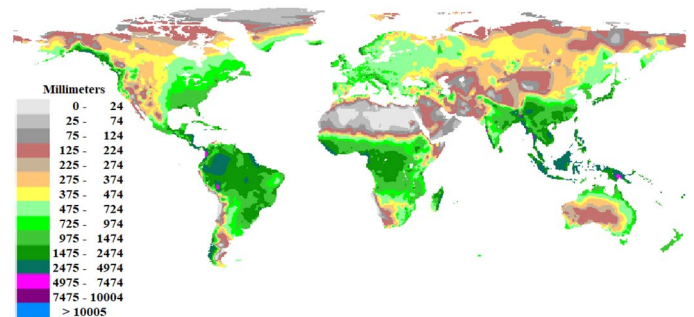


Fig. 1. Average annual rainfall across the world

Some studies that used shorter sample periods (days or hours), such as [3, 5], also tried to predict rainfall ahead for several steps. In this way, however, their modeling became more demanding, because the relevant physical variables, such as atmospheric pressure, temperature, wind speed, etc. were not known in advance and had to be predicted as well. The hybrid method in [5] relied on multilayer neural net of perceptrons with support vector regression and fuzzy C-means clustering for daily rainfall modeling in China. Paper [3] also modelled daily values, but with deep learning approach; its neural network contained an auto-encoder and a feedforward multilayer subnet.

Short-term prediction for 1-6 hours was attempted in [4] for Europe (Italy) by Auto-Regressive-Moving-Average, Artificial Neural Network and nonparametric nearest neighbours methods. In 2012, S. Yeon et al. defined an approach [11] for hourly prediction. Their model contained classification and regression tree (CART) and used temperature, wind direction, humidity and pressure as input attributes.

However, very few of these approaches [16] have been applied to the above-mentioned task of rainfall estimation, which is essential for meteorologists as well. One of these is the very short-term prediction (for 15-minutes) analysed in [10] for Australian data. That model contained 3 feedforward multilayer neural networks.

Another important factor influencing this problem is the geographical location where the precipitation is modeled. In the case of tropical or monsoon areas with very frequent rainfall, the rainfall modeling success is significantly higher due to the high rainfall regularity [9] [5] [3]. In the case of temperate zones, on the contrary, precipitation activity is significantly less frequent and less regular, bringing the highly stochastic nature of the task to the forefront [4].

In our paper we deal with rainfall estimation in the temperate zone of Central Europe (specifically, of Slovakia), considering also the question whether it is advantageous to have one common model for several locations or to model each location by a dedicated and geographically more restricted model. We also investigate the impact of specific regression model types on model accuracy when modeling both smaller and larger areas.

III. USE CASE

In our experiments we used data from two sources: surface meteorological stations and a weather radar. Both were provided by SHMU (Slovak Hydro-Meteorological Institute) as part of mutual collaboration. The data cover south-western Slovakia from 15. 02. 2016 till 20. 04. 2019. The first, radar data subset consists of reflectivity values measured periodically every five minutes. The width of the radar beam is 1 degree and within each five-minute period it scans full 360-degree circle for 12 predefined elevation angles (of which we actually used only 10). Its effective reach is 240 km divided into 960 longitudinal bins, each 250 meters long. For each bin, one value of radar reflectivity is returned during each 5-minute cycle. In the second subset, surface meteorological stations provide hourly data consisting of hourly precipitation and average values of surface atmospheric pressure, relative humidity, temperature and wind speed and direction (the latter two measured 10 m above the surface). All these data come stamped with the date and time of their measurement and the geographical coordinates of the station.

During the covered period of roughly three years, each surface measuring station produced approximately 25 000 hourly data records (with due allowance for servicing and other types of down-time resulting in missing or invalid data). It is also important to note that a vast majority of these records show no precipitation, which is the most typical case for Central European climate.

Given the limited number of records showing precipitation, data from one measuring station would not suffice for proper machine learning. Trying to find the right balance empirically, we created two datasets. The first contains data from three measuring stations that are relatively close to each other and, as a result, share similar local geographies (Dataset3).

The second (Dataset10) contains data from ten meteorological stations (including those in Dataset3) which met the following criteria:

- they are not too close to or too far from the radar (their distance from the radar is between 50 km and 200 km)
- They provide hourly precipitation values
- At least 60% of their data records contain valid values for atmospheric pressure, humidity, temperature, wind speed and wind direction.

As a result, Dataset3 contains 73 658 hourly data records, while Dataset10 contains 249 171 records covering a wider geographical area. Dataset10 is then bigger and might therefore permit building more representative and, consequently, more accurate predictive models. On the other hand, its data cover a wider range of measuring stations with different local geographies and other conditions influencing local weather and precipitation. It is therefore an open question which of the two datasets will enable us to produce more accurate predictive models.

IV. DATA INTEGRATION

Data integration consisted mainly in synchronisation of the five-minute radar data with the hourly measurements by the surface stations. This was achieved by averaging the 5-minute radar data into hourly records of average reflectivity for each bin covered by the radar. Afterwards we selected from these averaged hourly records those that were spatially close to the measuring stations, at several different altitudes. From these we tried to estimate the radar reflectivity at specific “significant points” at predefined altitudes near each measuring station, which we believed to be contributing most significantly towards the recorded precipitation levels. In determining the position of each “significant point” we took into consideration the wind speed and direction, as well as the (approximately) known terminal velocities of raindrops [13, 7, 8]. Our set of “significant points” also included those directly above each measuring station, as a kind of baseline. Given that our radar data accurately represented only discrete spatial positions (more specifically, bin centres), in estimating the radar reflectivity at each “significant point” we had to interpolate its value on the basis of two closest bin centres.

The situation regarding the altitude of radar beam above the sea level is shown in Fig. 2. From the known GPS position of the radar located at altitude h_a above sea level a radar beam is emanated under elevation angle β . The beam trajectory, shown in red in Fig. 2 is nonlinear, with gradually decreasing elevation angle.

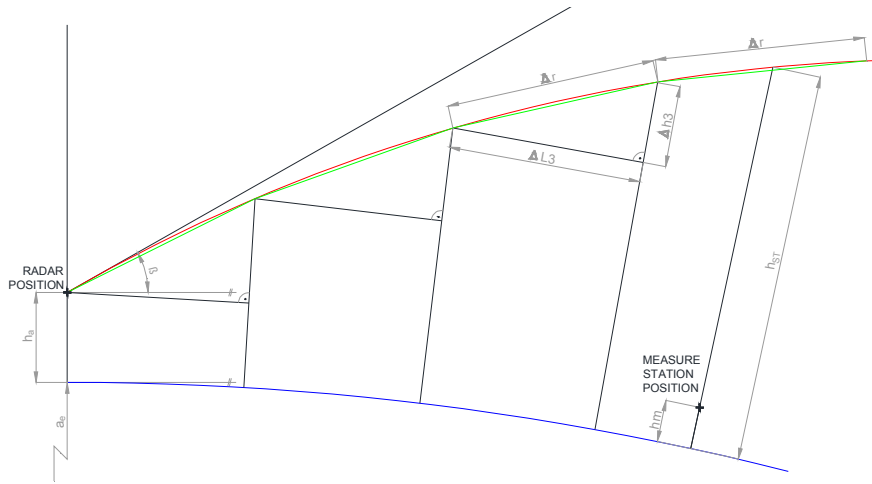


Fig. 2. Radar beam trajectory (in red color) in a physics-based model incorporating the curvature of the Earth

The dependence of the altitude of this beam on its trajectory r (travelled distance) is typically modeled by function (1).

$$H(r) = \sqrt{r^2 + (k_e a_e)^2 + 2r \cdot k_e a_e \sin(\beta)} - k_e a_e + h_a \quad (1)$$

This function represents the model from [6] reflecting the physics-based deflection of the beam (due to the thinning of the atmosphere with altitude) as well as the curvature of the Earth for longer distances. The meaning of the symbols in the equation is as follows:

- $k_e = 4/3$ (standard refraction coefficient)
- $a_e = 6371$ (Earth radius in km)
- $H(r)$ – beam height function, for a given travelled distance r
- β – elevation angle of the radar beam
- r – travelled distance of the radar beam in km
- h_a – altitude of the radar above the sea level in km
- h_m – altitude of the measuring station above sea level in km

The blue curve in the figure represents the sea level or zero altitude (curvatures of the beam trajectory and of the Earth have been exaggerated for the sake of clarity). For further processing we also need to know the travelled surface distance L (i.e. the projection of the curved beam trajectory r onto the idealized Earth's surface. For this purpose we have linearized the beam trajectory into short 50-meter fragments (shown in green in the figure), and within each fragment we consider the beam trajectory to be a straight line. This leads to the formation of a series of triangles for which we can calculate, first, the height increment Δh_i for each trajectory fragment through formula (2).

$$\Delta h_i = H(i \cdot \Delta r) - H((i-1) \cdot \Delta r) \quad (2)$$

Since the beam altitude increments over each trajectory fragment given by formula (2) are perpendicular to their corresponding surface distance increments ΔL_i , the length of the latter can be determined from the Pythagorean Theorem through formula (3).

$$\Delta L_i = \sqrt{\Delta r^2 - \Delta h_i^2} \quad (3)$$

The travelled surface distance L_k at the end of the k -th trajectory fragment could be approximated by formula (4).

$$L_k \cong \sum_{i=1}^k \Delta L_i \quad (4)$$

We have tabulated these values for each elevation angle β for quick and easy access. The access is actually inverted in the sense that we know the travelled surface distance L (on the basis of GPS coordinates of each measuring station) and need to retrieve the corresponding h and r values from the table. We actually interpolate both from the two closest values recorded in the table.

Regarding hourly data from surface measuring stations, we included in the integrated data the values of atmospheric pressure, temperature, humidity, precipitation (target attribute), wind speed and wind direction. We also added relative and absolute changes of pressure and temperature as new derived attributes.

Missing values for atmospheric pressure, temperature and humidity were replaced with those from the previous hour or, if those were missing too, by the average of all correctly measured values of a given variable. Missing values for wind speed and direction were replaced by those from the previous hour too, but if those were missing too, then they were set to zero. We only replaced missing reflectivity values in radar data if there were less than 8 of them in one data record, otherwise the incomplete record was entirely removed from our dataset. Missing reflectivity at a given altitude was replaced by the value from the closest altitude level above it multiplied by 0.8. Missing reflectivity values at higher altitudes were set to zero. All input attributes for machine learning are listed in Tab. I.

Apart from these attributes, our integrated data also contained date-time stamps and Ids of measuring stations for each data record, but these were not included in the training process of our machine learning models, because our initial experiments determined that they did not contribute positively towards prediction accuracy. At present we therefore rely purely on relevant meteorological variables in trying to find the right types of models that would accurately capture the dynamics of this highly stochastic system. In the future experiments we plan to include even the date-time stamps and station Ids in the training process, along with additional meteorological attributes provided freely by the NOAA (the US National Oceanic and Atmospheric Administration), or ECMWF (the European Centre for Medium-Range Weather Forecasts) such as wind speed and direction at different altitudes (which we believe to be very significant for accuracy).

All the attributes in Tab. I were subsequently normalised and their significance for real (recorded) hourly precipitation was then evaluated by various metrics (Information Gain, Information Gain Ratio, Chi^2 , Correlation coefficient) as shown in Tab. II.

TABLE I. OVERVIEW OF INPUT ATTRIBUTES FOR PREDICTIVE MODELS OF PRECIPITATION

Attribute Name	Attribute Description
r_prN	average radar reflectivity directly above the measuring station over the period of one hour for beam emitted under the N-th elevation angle (we only used the lower ten elevation angles: $N = 1, 2, \dots, 10$).
r_prXm	average radar reflectivity directly above the measuring station over the period of one hour at the altitude of X^* meters above the sea level. $X^* \in \{1100, 1350, 1600, 1850, 2100, 2350, 2600, 2850, 3100, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000\}$
r_pr_wXm	average radar reflectivity over the period of one hour at the altitude of X^* meters above the sea level at points located upwind from the measuring station; their precise locations reflect the wind speed, wind direction and the estimated terminal velocity of raindrops.
rdif_temp	relative change of air temperature over one hour (in kelvins)
dif_temp	absolute change of air temperature over one hour (in kelvins)
wind_angle	wind direction at the beginning of each hour
wind_speed	wind speed at the beginning of each hour
atm_humidity	relative humidity at the beginning of each hour
dif_pres	absolute change of atmospheric pressure over one hour
rdif_pres	relative change of atmospheric pressure over one hour
atm_pressure	average value of atmospheric pressure over one hour
atm_temp	average value of temperature over one hour

TABLE II. ATTRIBUTE SIGNIFICANCE LIST

Attribute Name	Information Gain	Information Gain Ratio	Chi^2	Correl. Coefficient
r_pr_w8000m	0.032115	0.064913	18944.1	0.402403
r_pr_w7500m	0.043844	0.069938	25308.2	0.408012
r_pr_w7000m	0.053881	0.078418	30958.9	0.417623
r_pr_w6500m	0.061658	0.075152	35533.3	0.421249
r_pr_w6000m	0.080335	0.089213	44780.6	0.425620
r_pr_w5500m	0.093785	0.093236	52787.4	0.435185
r_pr_w5000m	0.105812	0.095469	59357.6	0.432673
r_pr_w4500m	0.127224	0.110094	69729.9	0.432124
r_pr_w4000m	0.138292	0.113936	76872.6	0.430539
r_pr_w3500m	0.160041	0.122810	87905.8	0.433509
r_pr_w3100m	0.172025	0.135410	95220.8	0.434609
r_pr_w2850m	0.178411	0.135747	99694.1	0.435504
r_pr_w2600m	0.186254	0.138429	104344.6	0.438394
r_pr_w2350m	0.193477	0.147732	109334.1	0.445708
r_pr_w2100m	0.199224	0.151039	113812.0	0.451286
r_pr_w1850m	0.203589	0.160938	116997.8	0.452357

Attribute Name	Information Gain	Information Gain Ratio	Chi^2	Correl. Coefficient
r_pr_w1600m	0.205178	0.166894	118350.0	0.450619
r_pr_w1350m	0.206977	0.167562	120131.6	0.452749
r_pr_w1100m	0.209521	0.183722	122369.5	0.458436
r_pr8000m	0.031952	0.061331	19189.9	0.410524
r_pr7500m	0.044645	0.061938	25882.4	0.415021
r_pr7000m	0.054363	0.068275	31578.0	0.422264
r_pr6500m	0.062323	0.082548	36366.1	0.429003
r_pr6000m	0.085354	0.090123	47285.0	0.435761
r_pr5500m	0.096524	0.097599	54316.1	0.445783
r_pr5000m	0.109318	0.097066	60984.2	0.438913
r_pr4500m	0.129018	0.108068	70470.8	0.436057
r_pr4000m	0.138912	0.111952	77505.6	0.435595
r_pr3500m	0.160473	0.124941	88439.2	0.437331
r_pr3100m	0.172577	0.136404	95591.5	0.438023
r_pr2850m	0.178833	0.136721	100016.0	0.438708
r_pr2600m	0.186706	0.144716	104714.5	0.441191
r_pr2350m	0.193648	0.148955	109376.6	0.447582
r_pr2100m	0.199189	0.162865	113724.0	0.452354
r_pr1850m	0.203523	0.163692	116935.6	0.452846
r_pr1600m	0.204808	0.167660	118101.4	0.451447
r_pr1350m	0.206920	0.173161	120115.8	0.453724
r_pr1100m	0.209711	0.179136	122553.7	0.45979
r_pr10	0.001939	0.087460	1254.7	0.201707
r_pr9	0.012168	0.029485	7179.0	0.221831
r_pr8	0.032311	0.057258	19392.5	0.263456
r_pr7	0.048966	0.078048	29451.5	0.300231
r_pr6	0.066895	0.089385	40182.8	0.332857
r_pr5	0.097101	0.107272	57802.5	0.376547
r_pr4	0.129643	0.130380	76267.0	0.410029
r_pr3	0.167077	0.157753	97112.2	0.435517
r_pr2	0.202110	0.186347	118117.4	0.465551
r_pr1	0.196528	0.214759	119272.2	0.491772
atm_humidity	0.065358	0.018375	20762.0	0.122202
rdif_pres	0.007546	0.001995	2868.7	0.010890
dif_pres	0.006762	0.001696	2548.1	0.010820
rdif_temp	0.007083	0.002182	2339.1	0.067250
dif_temp	0.006109	0.001844	2050.9	0.069330
atm_temp	0.021225	0.007086	5942.1	0.002329
atm_pressure	0.015644	0.004489	5347.8	0.056360
wind_angle	0.002324	0.000821	802.5	0.003742
wind_speed	0.000517	0.000442	186.1	0.000694

Since it was not readily apparent from Tab. II which subset of input attributes would be optimal for model training, we created three versions and trained our models separately for each.

The first version contained all the 57 attributes listed in Tab. II.

In the second version with 54 attributes we omitted `wind_speed` and `wind_angle` (both measured at 10 meters above the ground level) which all our metrics pinpointed as relatively insignificant. One contributing factor to this might be that the estimated wind speed and direction at different altitudes were already taken into account in constructing the group of attributes `r_pr_wXm` for a predefined set of altitudes `X`. For similar reasons we removed atmospheric pressure (`atm_pressure`), once we could see that it did not help our Random Forest and M5P Regression Tree models during training on Dataset3 (Dataset3 speeded up the training considerably compared to Dataset10).

The third version contained only the 49 most significant attributes, specifically those with correlation coefficient above 0.10, χ^2 above 10 000, or information gain above 0.03. All these conditions pinpointed nearly identical subsets of input attributes, which we took as a sign of their reliability and suitability for our purpose.

V. MODELING

To model precipitation, we used several different types of models including regression trees, lazy methods, linear regression with kernels, SVM regression, etc. The two top-performing models were Random Forest (RF) and Neural Network (NN), and these will be our focus here because they significantly outperformed the rest. For these two types of models we subsequently varied their parameters and training methods in order to minimize the RMSE (Root Mean Squared Error).

Our RF model consisted of 100 individual tree models, while the NN model was a multilayer feedforward neural network of perceptrons, with 2 hidden layers (with 6 and 5 neurons, respectively) and logsig activation function.

Our neural network was trained by gradient descent with momentum and adaptive learning rate backpropagation [1, 2]. The training was stopped if there was no model improvement after 80 000 iterations.

Table III. shows average accuracy (represented by average correlation coefficient) of hold-out validation repeated with five different seeds (values 1 to 5) for both RF and NN models. Each seed pseudo-randomly defines different initial conditions of NN, and different subset of attributes in RF model. The accuracy is measured by the correlation coefficient of the predicted value against the real (recorded) value of hourly rainfall. The table shows the dependence of the accuracy on the number of used input attributes as well as on the dataset.

Dataset3 contains data from three weather measuring stations, while Dataset10 covers ten stations including the three from Dataset3. Dataset 3 is thus a subset of Dataset10. In our machine learning experiments with both datasets, the training set always included 60% of data records, while the testing set contained another 20% and the validation set the remaining 20% of data records.

TABLE III. COMPARISON OF ACHIEVED ACCURACY FOR OUR RF AND NN MODELS WITH VARIABLE NUMBER OF INPUT ATTRIBUTES. TO MEASURE ACCURACY, WE USED CORRELATION COEFFICIENT.

Model Type	Number of used input attributes	Model accuracy for Dataset3	Model accuracy for Dataset10
Random forest	57	0.7038	0.7202
Random forest	54	0.7033	0.7131
Random forest	49	0.6958	0.7078
Neural network	57	0.6993	0.6863
Neural network	54	0.7119	0.6986
Neural network	49	0.7101	0.6874

DISCUSSION

We can see that RF models trained on Dataset10 are more accurate than those trained on Dataset3. In this case, the benefit of larger dataset outweighed the potential negative impact of different physical conditions in the different station locations (local geography).

Quite surprisingly, NN models showed the opposite trend: those trained on three stations (Dataset3) were more accurate than those trained on all ten of them (Dataset10). This particular type of NN model (multilayer feedforward network of perceptrons with 2 hidden layers) thus seems to be more sensitive to the variations of local geography, which not even the additional data could outweigh.

Accuracy of our models also showed dependence on the number of input attributes. As we gradually took away the least relevant attributes, the accuracy of our RF models deteriorated. In this domain at least, then, RF models reach maximum accuracy for the full number of input attributes, which shows RF's robustness and tolerance of less informative attributes thanks to their prioritisation on the basis of information gain (which is typical of decision trees in general).

On the other hand, the greatest increase of accuracy for NN models was achieved by reducing the number of input attributes from 57 to 54. For Dataset3 and 54 input attributes we achieved the highest average correlation of 0.7119 and, within this group, one particular seed value produced the best performing model with correlation of 0.726. Reducing the input attributes further to 49 did not appreciably lower the average correlation: it stayed at 0.7101, which is quite close to the best average value of 0.7119. NN models are then more sensitive to the number of input attributes than RF, yet for Dataset3 and the reduced number of (the most significant) input attributes they significantly outperformed RF models.

Overall, our RF and NN models appear to be complementary, which may stem from their radically different internal structures and modes of operation. In the future it might therefore be worthwhile to combine them together through ensemble learning and check whether the average correlation would further improve.

In the meantime we accept the results reported here (with the maximum correlation of predictions with the reality of about 0,71) as the limit of achievable accuracy given the number of measuring stations from which we have data and

the maximum number of useful input attributes that we were able to define.

In the future this level of accuracy will serve for us as the baseline which we will try to improve upon, because to directly compare our results with other published results for different locations is problematic due to their different local geographies and climatic conditions.

Of course there also exist other NN types and structures, which might be able to further improve the accuracy even for our present datasets. We plan to apply some of them, including selected deep-learning models, in our future work.

At present we are trying to obtain new attributes capturing wind speed and direction at different altitudes. We focus on altitudes between 1 and 5 km, which appear to be the most significant contributors to the overall measured rainfall. To accurately determine the angle at which precipitation falls on each measuring station might therefore help us to pinpoint more precisely those weather radar bins at each altitude that are most likely to contribute to the measurement. This is currently our top candidate for further improvement of the accuracy of our rainfall estimation.

CONCLUSIONS

In this paper we have used machine learning to model and estimate rainfall, a distinctly stochastic variable, in the geographical region of Central Europe, specifically Slovakia. We have found out that two types of models (random forest and multilayer feedforward network of perceptrons) significantly outperformed other models with respect to overall accuracy measured by the correlation coefficient between the predicted and the recorded rainfall. At the same time we have seen that these two models were mutually complementary: Random forest appeared more suitable for global modelling comprising a larger number of different locations, while our feedforward neural network was more successful at providing more specific models focused on a smaller area or working with a smaller number of the most relevant input attributes. Random forest also proved more robust in the face of a larger number of input attributes, including some with low significance.

Comparing our results to those reported in [4], where hourly precipitation was predicted with correlation of about 0.69 – 0.70, our approach with radar reflectivity achieved correlation of 0.71.

In the future we plan to implement several modifications that could significantly improve prediction accuracy for this task. Specifically, we plan to add new attributes accurately reflecting wind speed and direction at different altitudes near each measuring station, and also to apply methods of ensemble learning and deep learning. Since random forest already is an ensemble learning method, combining it with further methods could be considered a kind of ensemble learning of the second order.

ACKNOWLEDGMENT

This research was supported by projects VEGA 2/0167/16 and U-COMP APVV-17-0619. We would also like to thank the Slovak Hydrometeorological Institute (SHMU -

<http://www.shmu.sk/en/?page=1>) for providing meteorological data as well as professional consultations.

REFERENCES

- [1] Chien-Cheng Yu, Bin-Da Liu: A Backpropagation algorithm with adaptive learning rate and momentum coefficient, *Neural Networks* 2002
- [2] M. Moreira, E. Fiesler: *Neural Networks with Adaptive Learning Rate and Momentum Terms*, October 1995, avail. 12. 08. 2019: <https://publications.idiap.ch/downloads/reports/1995/95-04.pdf>
- [3] Hernández, E., Sanchez-Anguix, V., Julian, V., Palanca, J., & Duque, N. (2016, April). Rainfall prediction: A deep learning approach. In *International Conference on Hybrid Artificial Intelligence Systems*, pp. 151-162, Springer, Cham, Avail. 01. 09. 2019: https://www.researchgate.net/publication/301320757_Rainfall_Prediction_A_Deep_Learning_Approach
- [4] Toth, E., Brath, A., & Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of hydrology*, 239(1-4), pp. 132-147, avail. 01. 09. 2019: <https://www.sciencedirect.com/science/article/pii/S0022169400003449>
- [5] Chau, K. W., & Wu, C. L. (2010). A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *Journal of Hydroinformatics*, 12(4), pp. 458-473, avail. 01. 09. 2019: <https://iwaponline.com/jh/article/12/4/458/3047/A-hybrid-model-coupled-with-singular-spectrum>
- [6] Joan Bech, Bernat Codina, Jeroni Lorente, David Beggington: The Sensitivity of Single Polarization Weather Radar Beam Blockage Correction to Variability in the Vertical Refractivity Gradient, avail. 01. 09. 2019: <https://journals.ametsoc.org/doi/full/10.1175/1520-0426%282003%29020%3C0845%3ATSOSPW%3E2.0.CO%3B2>
- [7] Van Boxel, John H., et al. Numerical model for the fall speed of rain drops in a rain fall simulator. In: *Workshop on wind and water erosion*. 1997, pp. 77-85.
- [8] Bringi, V., Thurai, M., Baumgardner, D. (2018). Raindrop fall velocities from an optical array probe and 2-D video disdrometer. *Atmospheric Measurement Techniques*, 11(3), pp. 1377-1384.
- [9] Duong Tran Anh, Thanh Duc Dang, Song Pham Van: Improved Rainfall Prediction Using Combined Pre-Processing Methods and Feed-Forward Neural Network, 2019
- [10] Luk, K. C., Ball, J. E., & Sharma, A. (2000). A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. *Journal of Hydrology*, 227(1-4), pp. 56-65.
- [11] Soo-Yeon Ji, Sharad Sharma, Byunggu Yu, Dong Hyun Jeong, "Designing a Rule-Based Hourly Rainfall Prediction Model", *IEEE IRI 2012*, August – 2012.
- [12] Aakash Parmar, Kinjal Mistree, Mithila Sompura: Machine Learning Techniques For Rainfall Prediction: A Review, avail. 01. 09. 2019: https://www.researchgate.net/publication/319503839_Machine_Learning_Techniques_For_Rainfall_Prediction_A_Review
- [13] Pieter Hazenberg: *Rainfall estimation for hydrology using volumetric weather radar*, ISBN 978-94-6173-630-7, 2013, avail. 16. 10. 2019: <https://edepot.wur.nl/260839>
- [14] S. Thorndahl, T. Einfalt, P. Willems, et al.: Weather Radar Rainfall Data in Urban Hydrology, *Hydrology and Earth System Sciences* 21, 2017, pp. 1359 – 1380, avail. 16. 10. 2019: <https://www.hydrol-earth-syst-sci.net/21/1359/2017/hess-21-1359-2017.pdf>
- [15] E. Morin, W. F. Krajewski, D. C. Goodrich, et al.: Estimating Rainfall Intensities from Weather Radar Data: The Scale-Dependency Problem, *Journal of Hydrometeorology*, 2003, volume 4, pp. 782 – 797, avail. 16. 10. 2019: <https://pdfs.semanticscholar.org/e704/9e65ef1cdca420c5781dc41fecb39fb53c28.pdf>
- [16] V. Chandrasekar, R. Cifelli: Concepts and Principles of Rainfall Estimation from Radar: Multi Sensor Environment and Data Fusion, *Indian Journal of Radio & Space Physics*, Vol. 41, 2012, pp. 389 – 402, avail. 16. 10. 2019: <http://nopr.niscair.res.in/bitstream/123456789/14747/1/IJRSP%2041%284%29%20389-402.pdf>

Building Neural Networks as Dataflow Graphs

Gabor Kruppai

Faculty of Informatics, Eötvös Loránd University

Budapest, Hungary

a180fm@inf.elte.hu

Attila Kiss

Department of Information Systems

Faculty of Informatics, Eötvös Loránd University

Budapest, Hungary

kiss@inf.elte.hu

Abstract—Generally, data processing programs can be organized into a data flow graph that defines the operations to be performed sequentially on the data. The operation of neural networks can also be interpreted in a similar way, in which the input data to be processed is a specific data set and the operations to be performed on the data are the layers of the net. Due to architectural reasons, the entire neural network graph must be built before actual running, thus it is necessary to change data flows' topological execution to evaluation preceding graph building since knowing the layers separately is not enough to operate the nets. As a solution for displaying editable program graphs, we created a framework in which data processing related Python packages can be described and the programs built from them can be visualized and executed.

Index Terms—neural network, user interface, graphs

I. INTRODUCTION

The main goal of the research was to help to create and run models as quickly, transparently and easily as possible when creating and testing models related to data processing. Our approach was to try to create a framework that can (mostly) automatically generate a graphical interface for a arbitrary Python packages which can be used to construct computational models related to the chosen package. As an example implementation, we choose Keras [2], which contains high-level APIs for neural networks, where the visual representation of structures could help during the development.

Visualization tools are already available for similar tasks (such as TensorBoard [3]), but they are mostly processing program codes and made for displaying purposes only, so the programs cannot be modified with these tools. Program-graph editors also exist, for example Node-RED [4] for IOT (i.e. Internet of Things) devices and RapidMiner [5] for data science projects in Java. Although solutions have already been made, they differ (from Python) in programming language and their graph nodes have to be prepared one by one without any automatization.

II. DATA FLOW GRAPHS IN PROGRAMMING

Stream-like structures [1] are most commonly used in applications related to data processing. In such programs, there are two main types of components: data transformations and data pipes. These components can be represented in graphs, where transformation functions can be interpreted as nodes and data pipes as the edges.

The project has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

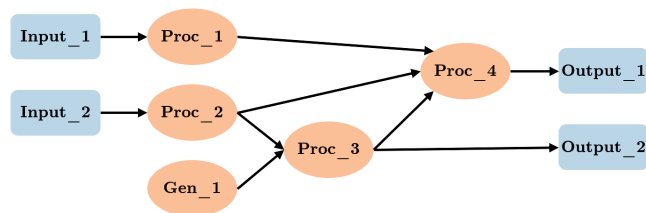


Fig. 1. Schematic diagram of a data flow graph with multiple input and output allowing edge splitting and joining.

Directed edges have only one purpose, to transfer the computed/processed data from one node to the proper input of another. The function that the node represents are only able to process or transform the data if all the required parameters are completely provided so that all the input data "arrived" from the preceding nodes in the "pipes". Finally, the result(s) are published to the output(s) of the node and the execution will continue analogously.

It is indispensable to the program termination that the graph does not contain directed cycles. In such case, a topological order exists and can be computed. Evaluating the nodes in that order results that the current node can get all its inputs immediately as all the nodes have already been executed on which it depends.

III. NEURAL NETWORK STRUCTURES

In its internal structure, a neural network can be considered to perform linear algebraic operations. The outer structure of the net can be seen as a multi-parameter function which takes and returns multidimensional arrays with predefined sizes. The result calculated from the input data is determined by the complex internal structure, which can store up to tens of millions of internal parameters, depending on the application. During the execution, matrices and matrix functions stored in this internal structure are used to compute predefined sequences of operations.

During the training phase, the proper iterative adjustments of the internal variables ensure that the model can get as close as possible to the solution of the given task which is defined by the training data set providing input-output pairs. To sum up, the algorithm approximates the function represented by the net by using fixed input and output values. During evaluation phase, the role of the external data and internal parameters

are interchanged, so that the internal parameters determined during the training phase are used to calculate the result.

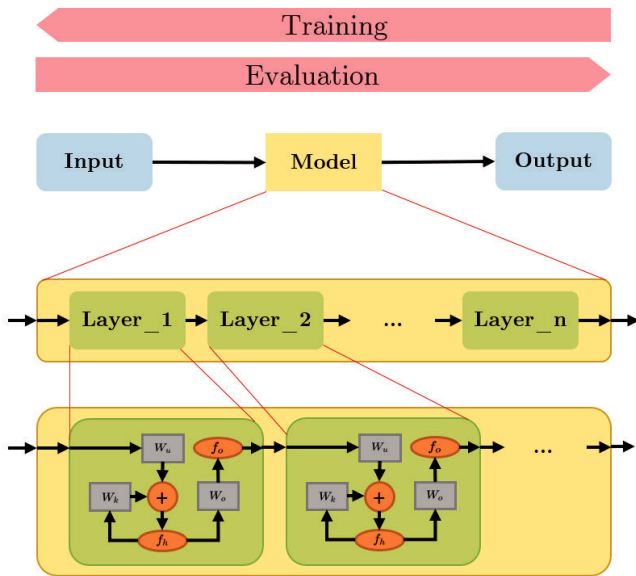


Fig. 2. Detailed example of neural networks' inner structure. A model consists of layers, a layer consists of linear algebraic operations and applied activation functions.

A. Neural Network Libraries

Deep learning software can be divided horizontally into low- and high-level libraries. The low-level, hardware related, linear algebraic and analytics libraries are optimized specially for basic operations and functions, the more complex structures are made by combining them. High-level libraries implement APIs that handle more complex neural layers rather than just variables. These two levels are strongly connected as they are built on each other, since neural layers implemented by higher-level APIs consist of a pre-compiled composition of variables and functions provided by low-level libraries, where the API hide its internal components to form a higher abstraction level.

The most widespread low-level library is the Google-developed TensorFlow [6]. Its high-level counterpart is Keras [2], which is written in Python, in which predefined layers can be combined as building blocks to create complete, trainable models.

B. Parallels with Data Flow Graphs

From both low and high-level approach, it can be seen that the data stream model could easily be applied to them if there were no need to modify the internal variables of the model during the training phase in respect of the training data set. With the exception of the training phase, the representation would be logically correct, even though the low-level implementation works differently. In addition, the full models have to be built first to train and run networks, and only then can be evaluated. It causes that we are not able to use and see the separated partial results of the neural model without building the whole.

Ignoring the underlying implementation and eliminating the initialization problem caused by the need of building the whole model, we get a logically correct, easy-to-read graph that could be edited easily to test and build new models without coding.

IV. TRANSFORM CALLABLE ENTITIES INTO NODES

Our goal is to reproduce existing libraries so that their elements can be graphically displayed and combined. This would lead to an interface where the functions and the classes which exist in the actual library can be used in a visual interface. In order to make it possible with larger packages, it is necessary to support the migration by automate the transformation between the Python modules and our entity description.

A. Python "inspect" Module

The Python "inspect" module provides wide variety of functions for retrieving information about modules, classes, methods, functions and other objects at run time. The most important aspect of the conversion mentioned above is to determine the existing elements of the modules and their attributes (e.g. type or signature). This could be achieved by reading the source code of the objects and analyze it.

As the main point is that we want to bind a graph node to a callable object, we need to know their signature, which is the name of the object and its parameters name, position, type, default value (if exists), and expected values. With the "inspect" module, not only these properties but even code comments and documentation can be retrieved if provided in the correct format. If we scan the module thematically, we can tell the path and the usage information for each object in the given module.

B. Python "importlib" Module

In Python, it is also possible to dynamically import any module in run time. The easiest way to do this - beyond the standard "import" keyword - is to use the built-in "importlib" package and its "import_module" function. If a specific objects location is known in its module (as a package string), it can be dynamically loaded into the memory in run time. It is exactly meeting our expectations when we would like to load a component only if it was used in the program.

C. Information Extraction

The first step in transforming nested package objects into graph nodes is to collect them into a list, in which the above-mentioned "inspect" module can help. It could be necessary to apply some filter criteria during the recursive package traversal, to skip undesirable objects and solve reference cycles. After the traversal, we will have a specific set of objects and their location in the tree-like modules.

In order to the ability to use the collected callable objects, the signatures and other calling settings must be determined for each object in the selected set. The signatures can be queried by the above mentioned "inspect.signature" function which will provide argument information extracted from the code text.

However, the information that obtained by the above method is still not sufficient to describe all the object calling styles. In order to achieve proper object calls with "type-correct" (i.e. classes that the function expect) arguments, additional call options and argument type information have to be gathered. If the default values are provided in the function's signature, the argument's type can be deduced, however, it may vary since Python are dynamically typed and there is a possibility to accept various types/classes (e.g. string and int). Unfortunately, accepted types cannot be predicted from the plain code text without having some kind of structured documentation, thus, this cannot be automated and requires human revision. Unfortunately, the call options, such as instantiation, application or optional type checks, also require manual completions:

- if the node represents a callable object, it has to be provided if the class should be instantiated or just passed as-is to the following nodes
- if a function call has side effects on an object, we have to consider deep-copy – if possible – the object before the function call to prevent future anomalies
- if an object has to be called multiple times in series by different nodes (application), we should indicate that in the node description wherein it is represented
- if the function has special argument types and we want to ensure the type correctness, we should change the automatic argument determination (as its working for basic types only)

There could be cases where intermediate calls or extra data/code transformations are necessary without the ability of calling an existing function from any package (no such function exists or special operation needed). To solve this problem, we created a custom module, called "fallback module" where all the user-defined functionality extensions can be placed. The advantage of the fallback module is to create and import utilities and wrapper functions in the framework without the need of installation.

After the automatic data extraction and the human revision performed, the following node (Python object) descriptor format will be available for each of the callable objects (we used JSON formatting to store the properties) to configure the graphical editor and inform the execution server about the usage of the objects:

```
{
  // Node name
  "name": "Input",

  // Object's location (fallback)
  // e.g. "keras.layers.Dense"
  // (here "kw" is self-defined
  // in the working directory)
  "package": "kw.models.Input",

  // "category" for visual styling

  // Input type(s) of the node
```

```
// (optional)
"itype": "Dataset",

// Output type of the node
// (optional)
"otype": "Layer",

// Instantiate or not
"instance": true,

// Call "previous" node with this args.
// (parameter application)
"apply": false,

// Deep-copy on pass
"copy": false,

// Maximum number of merged input(s)
// (max. number of preceding nodes)
"conns": 1,

// Argument descriptors
"args": [{

  // Name of the arg. in the code
  "arg": "shape",

  // Input type (shape ~ int tuple)
  // "nosupport" if not impl. visually
  "itype": "shape",

  // Required argument
  "required": true

  // "otype", "default", "conns", etc.
  // ...
}, {
  "arg": "name",
  "itype": "string",
  "default": null
}, {
  "arg": "dtype",
  "itype": "nosupport",
  "default": null
}, {
  "arg": "sparse",
  "itype": "bool",
  "default": false
},

// further arguments
// ...
]
}
```

V. GRAPH EDITING AND PROGRAM EXECUTION

A. Web Interface

To properly visualize the components of the processed package objects as nodes, the previously extracted node descriptions are used. In our solution, we decided to implement a browser-based program-graph editor with the help of an existing diagram editor library called "Go.js" [7], which was used with academic license in this work. The library's main profile is to provide templating and highly customizable diagram rendering.

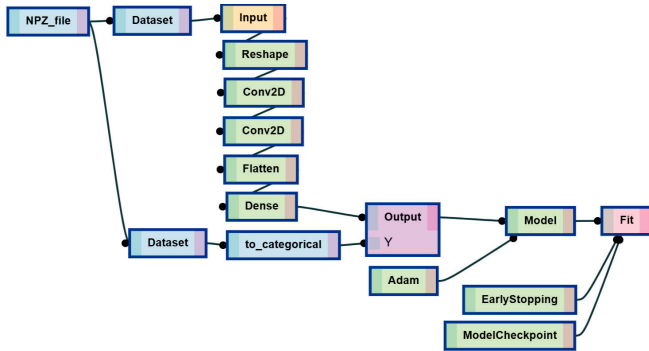


Fig. 3. The prepared functions can be dragged and dropped to the editing area and can be applied after each other by drawing edges (data pipes) between them. Every node can be displayed in "overview" or "detailed" mode (see Figure 4.)

Beside the calling instructions of the represented objects, the other important thing is to list the available argument fields. The arguments with basic typed default values are recognized during the automatic information extraction (IV-C) and associated with the arguments, however, it is still impossible to give a general solution which always determines all possible types. Even the predicted basic types can differ as Python is dynamically typed so they may accept other types as well as the predicted ones, so human revision is important cannot be omitted.

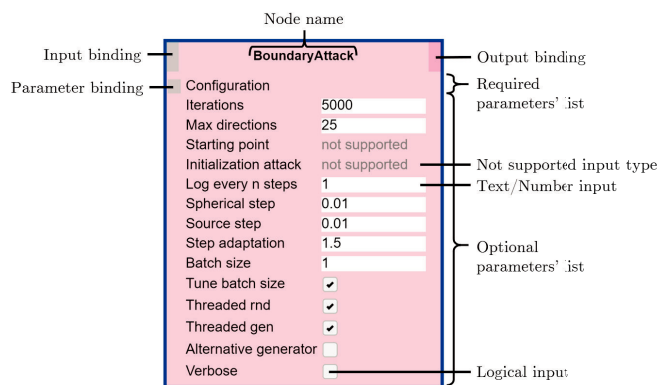


Fig. 4. Nodes can be generated from the configuration objects prepared based on IV-C. As some of the input types cannot be determined automatically, "not supported"* will be shown as they are not basic types but objects. *Unfortunately, it can only be eliminated or resolved by modifying the configurations by hand.

B. Saving and Submitting Models

In order to run a Python program from the graphical model in the browser, it has to be serialized and sent to a Python interpreter (on the server) where it can be parsed and executed based on the extracted serialized model. The visual models can also be exported into JSON format which is identical to the serialized model used by the model submission to the execution server. This JSON object contains all the necessary properties for both the execution server and the "Go.js" visualization, like node definitions and links as well as the nodes' actual arguments and execution state. Since this is the complete inner state of the client-side visualization, not only can it be exported, it can be easily imported as well.

C. Graph Execution

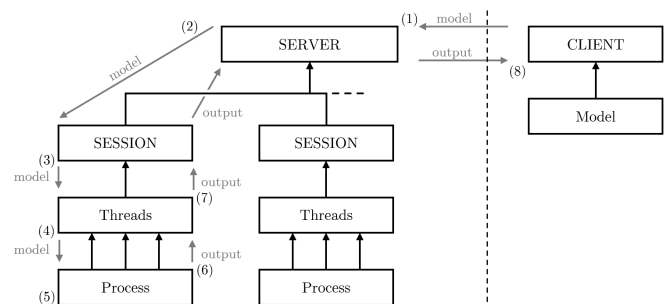


Fig. 5. Structure of the execution server. (1) Model serialization, sending to the server; (2) Execution session creation; (3) Spawn new threads for model execution; (4) Parse the received program and create topological evaluation order; (5) Evaluate the graph nodes in a dedicated process; (6) Capture standard outputs; (7) Notify the session about the progress; (8) Send back progress and output data.

To execute the client-edited program, we also implemented an execution server (in Python) where the serialized program model (i.e. graph) can be sent to, parsed and executed. As the server receives the executable model from the client, it parses the serialized data and rebuilds the same graph as it was created on client-side. It also checks if the received graph is cycle-free and then sorts the nodes into topological order to prepare them for sequential processing. The evaluation starts from the first node of the topological order and moves towards one by one, taking the previously computed values from preceding nodes – if they are necessary for the current node. During the evaluation of the current node, the server-side framework read the represented object's package path from the model graph and try to import it from the installed packages first. In case of the requested module or function does not exist globally, a user defined fallback module will be tried, which can be useful not only for error handling but for package extension. If even the fallback module and its function is missing, error will be thrown and the whole execution stops.

Another crucial point is the side effect and "noncopyable" object handling. If a node's output value is bounded to multiple following nodes but calling a function on it causes modification in its value, only the first node will get the original value and the others may fail because of the "unintended" value

change through references. To solve this problem, it should be ensured that the node’s original return value does not change by deep-copying – if possible – it on demand. To avoid most of the unexpected behaviors, the framework tries to make a deep-copy from all objects if possible, otherwise leaves them as they are.

Each evaluation step (evaluation of a single node) starts and ends with meta signals which are captured and forwarded to the client to inform the user about the progress and the time consumed. Beside the meta signals, standard outputs’ contents are also captured and sent back to the client.

VI. EXPERIMENTS

To properly test the developed framework, we built several neural network models with it. In order to manage the required data for the nets, we used exported ”numpy” objects (”.npz” files) to store and load data sets for both training and validation, so additional ”npz” data loader nodes were implemented to work with them. Showing the working mechanism and capabilities, a simple sequential (on Figure 6) and a functional (on Figure 7, with layer splitting and merging) model presented.

The nodes with different logical functionality are marked with different colors. On the screenshots below, blue indicates data managing (e.g. loaders, transformations), yellow and purple means model input and output, green nodes belong to the neural model (e.g. layers, optimizers, functions, configurations) and red indicates the compute-intensive calls (such as training and evaluation).

We made 91 nodes available from Keras package in our framework in total, from which 76 was auto-generated, 10 required a wrapper class and 5 needed to be redefined including the 3 data nodes. In addition, an other library – Foolbox [8] for generating adversarial attack inputs with wide variety of methods – was also transferred and tested with our program. Foolbox required a little more (10) functions to be wrapped in totally self-defined objects and 32 function needed almost identical one-lined wrapper functions which could be eliminated in the future by recognizing even more calling information. These manual extensions were enough to cover more than 45% of Keras and around 95% of Foolbox functions and classes.

A. Extending functionalities

Adding new packages and functions are also possible by generating (see IV-C) additional node (object) descriptors and appending them to the configuration file which contains these data. In our examples, we only ported specific parts from popular neural network related libraries, however, other fields also can benefit from the base framework. Our original goal was to make it possible to visually build programs with as little preparation overhead as possible. This technique could also help in data processing and transformation if the model is extended with the required functions.

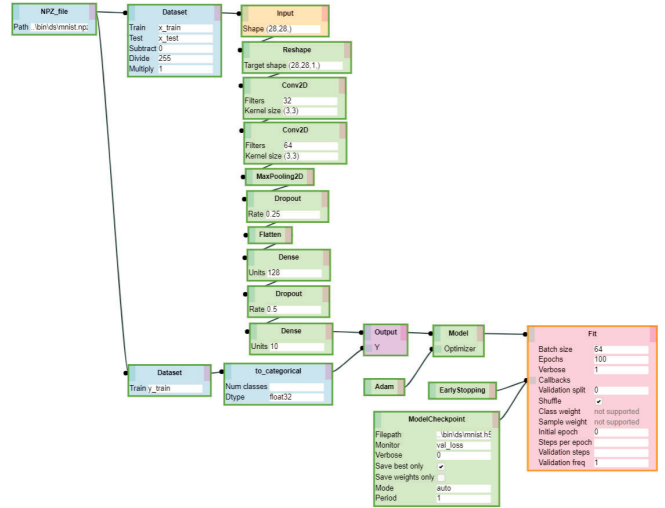


Fig. 6. Simple MNIST convolutional neural network sequential model from https://keras.io/examples/mnist_cnn

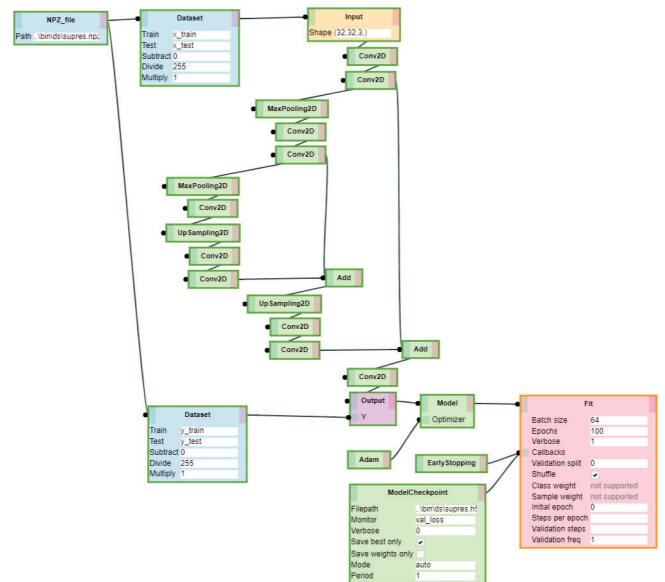


Fig. 7. Deep Denoising Super Resolution (DDSRCNN) from the collection at <https://github.com/titu1994/Image-Super-Resolution>, referring to [9]

VII. CONCLUSION

The data flow structure has restriction over general programming, namely with circular object dependencies and side effect handling. In statically typed languages, the function-node transformation would be feasible in contrast to dynamically typed languages where some of the nodes can be generated automatically but there is no fully general solution to handle every function signature. However, creating editable graph view for Python packages with a relatively small amount of work is quite useful. On the one hand, this tool can be helpful for professionals in data processing and testing tasks like multimedia (image, audio, video) manipulation pipelines

(e.g. with "ffmpeg"), data mining applications and cycle-free network modelling. On the other hand, it can be used in education too, for example to teach basic programming for children as no language and coding knowledge is required.

Possible improvements could be made if the execution system was changed from single node evaluations to actual program generation and running. That would bring a completely different backend structure and possibly increase the graph complexity on editing which is against the endeavor of simplification but worth to test in future works.

REFERENCES

- [1] Kodosky, J., MacCrisken, J., & Rymar, G. (1991). Visual programming using structured data flow. Proceedings 1991 IEEE Workshop on Visual Languages, 34-39.
- [2] Keras.io. (2019). Home - Keras Documentation. [online] Available at: <https://keras.io/> [Accessed 1 Sep. 2019].
- [3] TensorFlow. (2019). TensorBoard — TensorFlow. [online] Available at: <https://www.tensorflow.org/tensorboard> [Accessed 1 Sep. 2019].
- [4] Michael Blackstock and Rodger Lea. 2014. Toward a Distributed Data Flow Platform for the Web of Things (Distributed Node-RED). In Proceedings of the 5th International Workshop on Web of Things (WoT '14). ACM, New York, NY, USA, 34-39. DOI: <https://doi.org/10.1145/2684432.2684439>
- [5] RapidMiner. (2019). Lightning Fast Data Science Platform for Teams — RapidMiner. [online] Available at: <https://rapidminer.com/> [Accessed 1 Sep. 2019].
- [6] TensorFlow. (2019). TensorFlow. [online] Available at: <https://www.tensorflow.org/> [Accessed 1 Sep. 2019].
- [7] Farrukh Shahzad, Tarek R. Sheltami, Elhadi M. Shakshuki, Omar Shaikh, 2016, A Review of Latest Web Tools and Libraries for State-of-the-art Visualization, Procedia Computer Science, DOI: <https://doi.org/10.1016/j.procs.2016.09.017>.
- [8] Jonas Rauber, Wieland Brendel and Matthias Bethge. Foolbox: A Python toolbox to benchmark the robustness of machine learning models, 2017; arXiv:1707.04131.
- [9] Xiao-Jiao Mao, Chunhua Shen and Yu-Bin Yang. Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections, 2016; arXiv:1606.08921.

Efficiency of the relational database tuple access

Michal Kvet, Karol Matiaško

Department of Informatics, Faculty of Management Science and Informatics

University of Žilina

Žilina, Slovakia

Michal.Kvet@fri.uniza.sk

Abstract— Relational databases form the core part of the data management in current information systems. The number of data is still rising and the structure is more and more complicated and evolving. If the data are bordered by the time spectrum, problem is even deeper. This paper deals with the relational database system architecture and proposes own techniques for optimizing data location and access to the tuples to get relevant data effectively in a proper time. Own proposed solution is based on limiting the impact of the whole table scanning necessity. Thanks to that, the performance is significantly shifted and improved, whereas, in our solution, index approach can be always used, since each data row is delimited by the primary key definition.

Keywords—data access; data tuple; versioning; temporality; index management

I. INTRODUCTION

The amount of the data to be processed, managed and evaluated is significantly high and is still rising. Numerous data evolve over time and individual changes and versions need to be stored for the purposes of later further evaluations and analyses. Relational database technology was firstly proposed in the 60ties of the 20th century and it is still used these days, as well. The strength and power of it are associated with the robustness of the mathematical background forming it – relational algebra. Data in the relational databases are formed in the shape of individual tables and relationships between them. Each row is commonly delimited by the unique identification defined by the primary key, which creates unique index automatically, as well. Individual connections between tables are defined by the foreign keys referencing primary keys, respectively unique indexes. To access the data, particular data block must be loaded from the physical database to the instance memory, where the requested row is identified. For the access, various index methods can be used, if the corresponding index is defined and suitable. Many times, therefore, the data access is shifted to the necessity of all table blocks scanning. When the data fragmentation after the numerous data operations (*update*, *delete* statements) is present, the problem is even sharper.

The main contribution of this paper is associated with its own index technique, which removes the necessity to scan all the blocks to locate the row based on the conditions. This method, therefore, removes the impact of the whole table scanning necessity, even those blocks, which are actually

empty but associated with the table. The core of the system is based on the extension of previously proposed techniques – *Flower index approach (FIA)* and *Master index* [9] but moves the processing into block precision. It uses index just as the data block locators, thus the row identifier itself is shifted to the block precision granularity.

II. SYSTEM ARCHITECTURE

The database system consists of two main architectural structures. The database itself is represented by the physical store located in the disc file systems. Individual data tuples are located in the extents, each of them contains several data blocks, where the tuple is physically located. New storage space for the particular table is always allocated as the extent, not the single data block. As described later, it is a significant performance characteristic influencing the process of block scanning during data access operation. The extents that have already been associated with the table are permanent and, by default, unallocable, except in the event of a deep system and storage restructuring. The first extent is associated with the data segment consisting of the object definition itself. Individual extents are then connected via linked list. The last block of the last extent is pointed by the *High Water Mark Symbol (HWM)*. Thus, when all blocks are scanned, the operation starts with the first block associated with the initial extent of the object and ends with the block, to which *S* points.

The database structure is connected to the database instance during the mounting process of the system *startup*. The database instance is the second part of the architectural structure. It is delimited by the background processes managing the whole system and memory structures. The global description of the architecture and its enhancements can be found in [1] [2] [12]. The instance is a complex set of structures (mostly shared across the whole system forming *Shared Global Area – SGA*), which maintains and manages data and the system itself. *SGA* consists of descriptive data managed by *Shared pool* structure characterizing data dictionary and object structure definition, parameter settings, and description, as well as cache for recently executed code and stored result sets. It contains the *Log buffer* to manage change vectors and *Buffer cache*, as well.

For the data retrieval, the most important part is defined by the database *Buffer Cache*, which is modeled as the matrix of the blocks with the same size, as the database block itself has. When data are to be processed, individual blocks from the

physical databases must be loaded into the memory and such structure is the destination. Individual blocks can be either *occupied* or *free*. *Free* block can be directly used for new block, respectively block. If the block is *occupied*, it can be either *dirty* (new or changed data are present, which are part just of the log files, but have not been reflected in the database, yet) or *clean* (there is no change in the block in comparison with the physical database block). The transfer of the data from and to database is shaped in the block granularity, individual rows as the part of the block cannot be transferred, at all.

The architecture of the database system is shown in the fig. 1. Interconnection between instance and database is always operated by the background processes, user can never access the database directly. The user is represented by the user process on the *client site*. On the server, the *server process* is created by the *Process Monitor* background process activated by the listener.

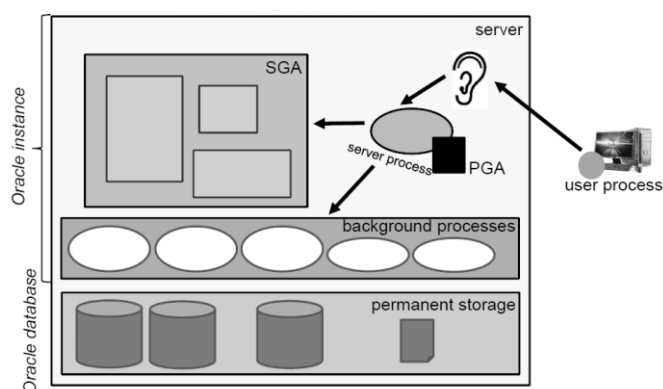


Figure 1. Database architecture [12]

Data access and tuple location inside the database can be done either directly by scanning all the blocks associated with the table (under *HWM*) or by using an index. Section 3 describes the existing index approaches and techniques used currently in the relational systems. To emphasize the problem and necessity to optimize the performance and limit sequential data scanning, it must be noticed, that the *HWM* symbol can never be shifted back to the left, thus the number of blocks (extents) associated with the table is still rising. Sequential scanning is therefore significantly demanding and proposes performance limitations, whereas many fragmentations and partially or even completely empty blocks can be present.

III. EXECUTION AND INDEXING

Data retrieval and access management form the crucial stage of the evaluation process. Statement execution is defined by the following steps – *parse*, *bind*, *execute* and *fetch*.

Parse operation is the first processing step, in which the system checks the existing plans for such query, access rules, privileges, etc. If so, stored version of the plan is in the *Library cache* of the memory instance. The aim of this phase is to select the best possible way to access the data. Thus, also the selection of the index is part of this processing step. If the execution plan is not available, optimizer decides, which index

will be used, respectively how to access data. The last possibility is correlated with sequential scanning. In the *bind* phase, variables are expanded into literals. Afterward, *execution* phase is started. During this phase, data are read from the *Buffer cache*, if the particular data are already there. By default, however, it is necessary to load the data from the database, to transfer the relevant data blocks into the memory clean blocks of the *Buffer cache*. Thus, system must find the blocks in the memory to be the data source destination. If they are not available, system notifies *Database Writer* background process, which stores some piece of the *Buffer cache* into the database resulting in transforming *dirty* block marks into *clean*. As evident, such activity lasts some time (influencing the performance parameter – total time of the processing), therefore the aim of the system is to ensure an appropriate amount of free blocks. Methods to ensure that are described in [12]. *Execute* step reflects the data access and query execution itself followed by the *Fetch* phase, where result set is composed and proposed to the user.

Fig. 2 shows the processing steps of the query evaluation. There is also a *preprocessing* phase defined by our previous research. The aim is to modify the shape of the command to cover existing parse forms, if possible. Principles are defined in [10].



Figure 2. Processing steps of the query evaluation

During the data evaluation and processing, a significant role plays the data access methods and technology for data location. Each primary key, respectively unique constraint definition automatically forces the system to define the index. In addition, individual developers create their own secondary indexes to optimize the system by locating data using *ROWID* values in the leaf nodes of the index. *ROWID* specification can slightly differ across the database systems, however, in principle, it consists of the object identifier and localization of the data by the pointer to the data file, data block a position of the row inside the block [6] [12].

From the architectural point of view, several index structures can be identified, from B-trees, B+trees, bitmaps, hash indexes, their variants and enhancements (index-organized tables, reverse key approaches, function-based, virtual, domain, clusters, etc.), up to web search and textual indexes [3] [4] [12] [13]. Currently, the most often used index structure is B+tree, which maintains efficiency despite a lot of data amount and multiple changes. Moreover, primary keys and automated index structure definitions are associated with such a technique, as well.

B+tree index consists of a balanced tree in which each path from the root to the leaf has the same length. In this structure, we distinguish three types of nodes - root, internal node, and leaf node. B+tree extends the concept of B-tree by chaining nodes at the leaf level, which allows faster data sorting [5] [6] [12]. The model of the B+tree is in fig. 3.

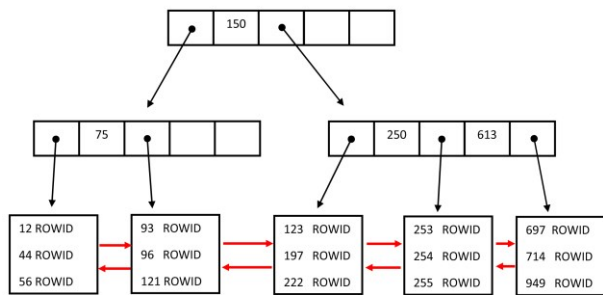


Figure 3. B+tree

Performance of the system related to the index reflected by the block size and tablespace definition is an important task, to be dealt with, as well. Principles and impacts can be found in [1] [12].

When dealing with the indexes, the main role plays the access method for obtaining data. Access path selection is one of the most important parts of the optimizer decision in *parse* phase. In principle, techniques can be divided into two categories. If the index is not used, scan of all blocks (rows) of the table is necessary to perform. If the table is small, there is no significant limitation. However, as the number of data increases, there is rapidly growing problems caused by fragmentation [3], disk access and migrated row issues [5] [14]. To remove these limitations, index structures are usually developed. It is, however, important to say, that it is not beneficial, even possible to develop all the indexes based on the queries, whereas the structure of them can evolve and be changed dynamically. Moreover, index management can be the bottleneck of the system, as well, whereas all changes (insert, update, delete) must be reflected the all of the indexes before approving transactions [7].

Access path using index usually consists of the combination of two methods. In the first phase, the particular *ROWID* value set is constructed reflecting the selected index. This method category is covered by the *Index Unique Scan* method, where no more than one row is selected (based on the conditions, which ensure uniqueness), *Index Range Scan* or by the group of *Full Scans*. In such a case, all index entries are scanned and evaluated to locate the data. It is generally more efficient in comparison with scanning the whole table, whereas the index has usually lower size and is located in the memory (at least partially). The specific index access method has been developed in Oracle 9i and is named *Index Skip Scan*. It can be associated with the composite indexes, where the optimizer skips the leading index column. Thus, database splits composite index into logical subindexes.

The index access method is always correlated with its suitability expressed mostly by the *WHERE* condition of the query. If the attributes, respectively functions occurred in such clauses of the *SELECT* statement are not part of the index, it is denoted as non-relevant for the query by the optimizer. Many times, therefore, optimizer decides to use sequential scanning with all its negatives.

The main contribution of this paper is associated with the proposed approach to remove the necessity to use *Table Access Full (TAF)* method to locate the data in the physical storage. Such method is characterized by the necessity to scan and access all data blocks created for the data table object. The main negative is just associated with the *HWM* symbol, where also empty blocks must be loaded into the memory, which significantly demands the throughput of the system and the whole performance. Proposed techniques use the indexes just as the data location pointers, thus, mostly the whole index is scanned.

IV. OWN ACCESS TECHNIQUES

Full Index Scans methods are based on the fact, that the selected index attributes are not in the suitable order, but the decision, whether such data row should be part of the result set or not, can be done on the leaf layer of the index. Thus, the index is mostly fully scanned, but the output of the method consists of only relevant *ROWID* values, which are consecutively processed in the second stage by the method *Table Access by Index Rowid (TAIR)*. This method is only accessing data based on the input pointers.

The principle of the proposed solution is different. The defined index is used as the data access layer to only relevant blocks. The index itself always shifts the processing to the block, where at least one data tuple is present. Apart from the migrated row, it is always true, whereas *ROWID* values are present in the B-tree index type as the locator to the data. Thus, if there is no pointer to the block, particular block is not evaluated and loaded, at all, whereas system is sure, that no data are located there. In this case, therefore, the solution scans sequentially all leaf nodes of the index, which are chained using double site linked list. After obtaining all *ROWID* values from the index and composing the output set, the second method is started – *Extraction*. The essence of this method is based on obtaining pointers to the data blocks. The management is not to access the individual records themselves, but whole blocks are processed at once. The reason is mostly reflected by the block transfer to the memory *Buffer cache*. As mentioned, it is not possible to transfer only part of the block. However, if just one row itself would be scanned, it could happen, that particular block would be removed from the memory by the *Database Writer* process to free space to other blocks of the other transactions. As a consequence, particular block would be necessary to be loaded into the memory more than once. Thus, the aim of this method is to remove the necessity to use the *TAF* method, which results in transferring empty, respectively irrelevant data blocks into the memory. The objective is to obtain blocks in which there is at least one record. The difference between *TAF* method and our proposed solution is shown in fig. 4. Let's have *six* data blocks formed by two extents. The *third* block and the last *sixth* are empty. In case of using *TAF* method, all data blocks under *HWM* are to be scanned – *six* blocks. Our approach removes blocks, which do not hold relevant data. Thus, the processing is managed by only *four* blocks. It is just an easy example, but in the complex environment, several

empty blocks and data fragmentation can be present as the result of data changes. Processing steps and evaluation principles are shown in fig. 4.

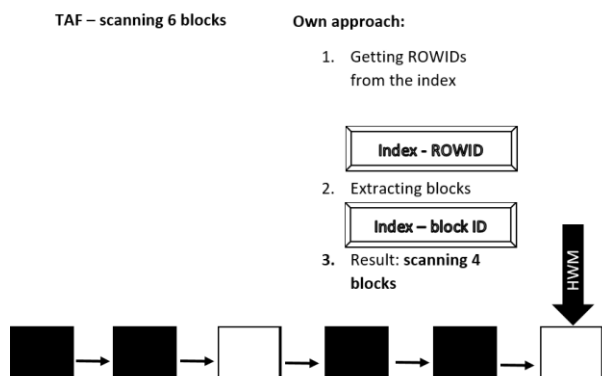


Figure 4. Evaluation steps

Obtaining *ROWIDs* from the index requires the decision, which index will be used for that. Our solution has only one strict requirement – pointers to all the data must be present there. In other words, index based on nullable data attributes cannot be used (in case of using *B+tree*, where *NULL* values are not present in the index). In principle, any index passing such condition can be used, however, the goal of the database optimizer extension is to choose the index that is the smallest, respectively its loading into memory would be the cheapest, expressed in total cost defined by the time and system resource consumption.

When dealing with the index, emphasis must be done also to the problem of migrated row. In this case, a particular data row is not in that block, where *ROWID* points, but to the other which is selected by the background process. Such a situation occurs, if the newly updated row does not fit the original size – data require more disc storage than available in the original block. In this case, naturally, data are shifted to another suitable block, but the index cannot detect such situation. As a consequence, no *ROWID* points to that new block. Thus, such block would not be processed and would be marked as empty (if no other data are present). Therefore, at first sight, proposed solution is not relevant, whereas it can ignore data composed during the migrated row problem management. Fortunately, the reliability is ensured as the result of the index definition itself. Each objective data block contains a pointer to another block if the original record has been moved. In our solution, processing stage itself has been extended to two phases. After the list of the processing blocks is created, individual blocks are gradually loaded into the memory and relevant data are retrieved, based on the *WHERE* condition of the original query (first phase). During the processing, the system may encounter a block that contains a pointer to another block, that is migrated. However, the newly-detected block is not processed immediately but is queued. In the second phase, newly defined queue is accessed and evaluated. Before loading a block, the system checks, whether such block has been processed before, or not. If so, it

is ignored. If not, transferring into the memory is executed followed by the evaluation of records stored in it. Even in this block, a migrated row can be detected. In such case, the detected block is placed at the end of the queue and the processing continues with the consecutive one. Our proposed solution is reflected by the data flow of the global evaluation in fig. 5.

V. RESULTS

The aim of the proposed technique is to extend the model of the database optimizer and provide methods, by which the sequential searches applied to all data blocks can be eliminated.

Computational study of the experiments is characterized by a sensorial temporal network environment. Experiment results were provided using Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production; PL/SQL Release 11.2.0.1.0 – Production. Parameters of the used computer are Processor Intel Xeon E5620; 2,4GHz (8 cores), 16 GB RAM, 500 GB HDD.

Data tuples and models come from the real data environment characterized by the 1000 sensors producing data ten times for one second. Some data portions are not always changed, respectively produce the same value. For such situations, data are not changed and the original state validity is extended (for these experiments, we use the following rule – if the change between consecutive values is less than 1%, values are considered as the same). Thus, the average amount of new values is approximately 1000 per second. The amount of data after one hour is 3 600 000. The data model is based on object granularity in the temporal spectrum [11].

To evaluate the performance, four environment characteristics were proposed. The first model (*model 1*) does not use any index. In that case, it is necessary to scan all data blocks by using the *TAF* method. The second model (*model 2*) is referential, in that case, the optimal index for the query is used. In the real environment, it is, however, very complicated to have all suitable indexes, mostly due to the demands during *Insert*, *Update* and *Delete* statements, whereas all indexes must be appended. The third model (*model 3*) consists of only one index, which is, however, not suitable – no leading attribute of the query is part of it. In this case, system is forced to use the index by query hints. The last model (*model 4*) is associated with proposed solution in this paper.

Reached results are shown in fig. 6, where the processing time is modeled. 20 percent of the total data amount is collected. Model 1 required 61,2 seconds for the processing, model 2 provides improvement at rate 78% (13,5s). In such case, optimal *B+tree* index is present in the system. The third model forces the system to use index, which is, however, not suitable for the query. As a result, processing lasts 69,8 seconds, which represents a deterioration of 12,3% in comparison with model 1. The proposed solution resulted in 14,1 seconds - slowdown less than 4,5% in comparison with the optimal index – model 2. It is, however, necessary to mention, that the proposed solution is universal and resistant to any index change.

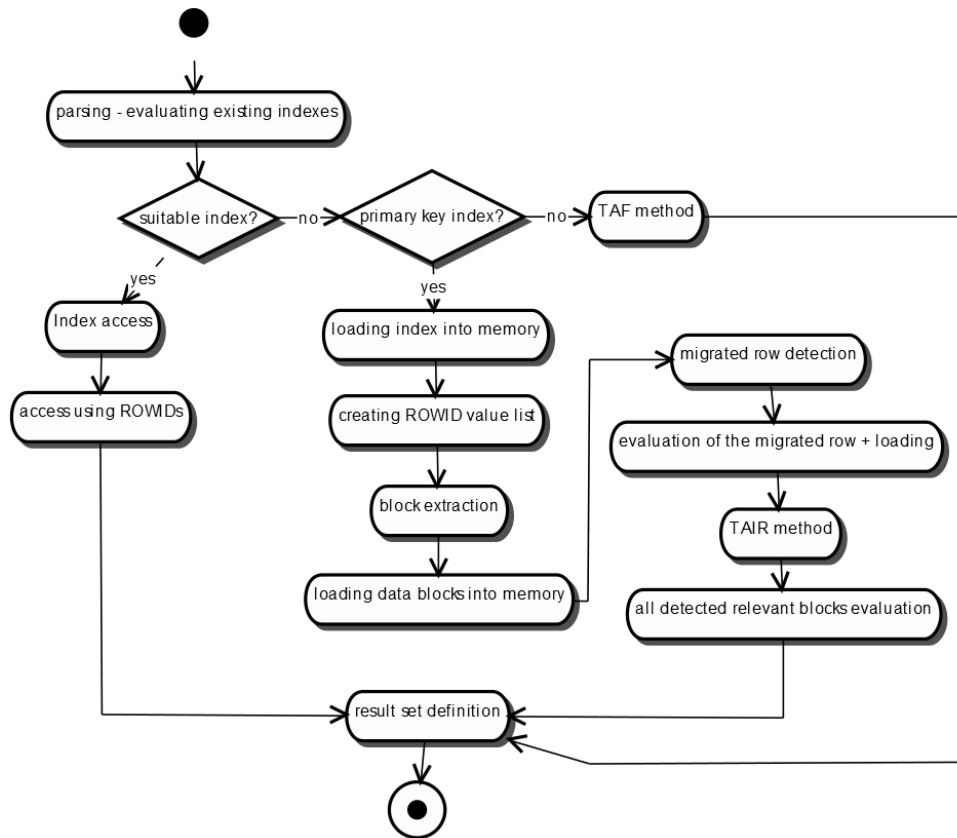


Figure 5. Evaluation of data flow – our proposed approach

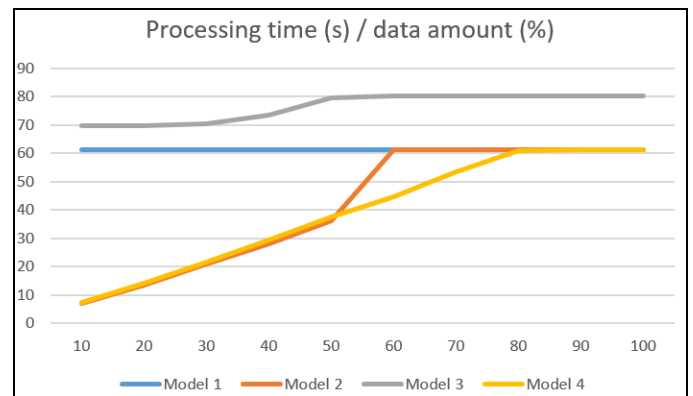
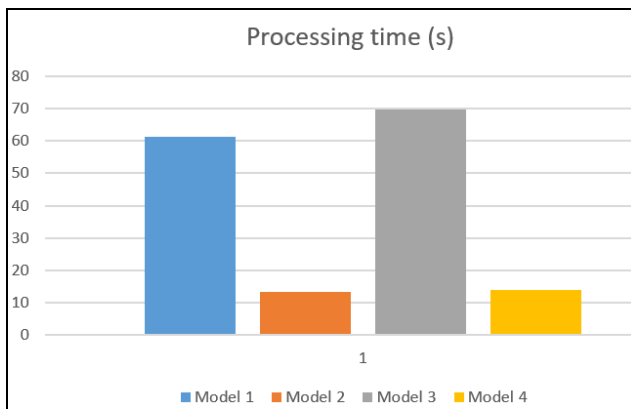


Figure 6. Performance results – processing time

Fig. 7 shows the results in terms of the relationship between the performance and amount of data to be retrieved (expressed in the percentage). When comparing *TAF method* (model 1) with proposed solution (model 4), it provides significantly better results, if the amount of required data correlated to the whole data set is lower than 80%. E.g., if the data amount is 40%, model 4 requires 29,4 seconds, model 1 needs 61,4 seconds. An optimal index environment is usable up to 50% of the whole data amount set and provides approximately 1 percentage improvement in comparison with proposed solution determined by model 4.

VI. CONCLUSIONS

Data access and global management of the block transfer between the database and instance plays a significant role in terms of performance. Reliability and usability are delimited by the efficiency of the data location. Each transfer is secured by the background processes to ensure transparency and limit the options for storing and modifying existing data with no correlation to the integrity rules. Process of the query evaluation consists of four steps – *parse*, *bind*, *execute* and *fetch*. Highlighting performance, the most significant aspect is delimited by index access method usage, which is selected in the *parse* phase. The index itself does not need to be suitable for the processing, whereas the order of attributes and its

structure does not fit the conditions of query. If there is issue related to the attribute order with emphasis on the query conditions, system can use *Full Index Scan* methods, where the entire index is searched, however individual evaluations are performed directly on the leaf level of the index, before the access to the data blocks in the physical storage. The principle of the proposed solution described in section 4 is different. Individual *ROWID* values are grouped together by shaping its definition to the block granularity. Thanks to that, each block is evaluated and transferred to the memory no more than once. The implemented solution covers significant benefits if no suitable index is used to overcome the fragmentation and size limits of the database object. Data blocks, which are empty, are not accessed at all, whereas no index pointer heads on them.

Transaction robustness and transparency are directly related to the log management covered by the *Log Ahead Rule*, where data changes must always be registered in the log before the execution of the particular change physically. Data block itself, however, does not need to contain the relevant data reflecting the query, whereas multiple transactions are present in the system. The particular data block may already contain changed data, but the query requests the original one, just because it was run before the change itself (or correlated to the transaction).

In the future, we would like to extend log management to remove the limitation of historical image composition caused by *Snapshot too old* exception in the non-temporal data environment. The principle will be based on splitting log files into transaction granularity with dynamical decision making of its future usage possibility. We would like to extend the log management to ensure more proper data with regards to the reliability, as well.

ACKNOWLEDGMENT

This publication is the result of the project implementation: Centre of excellence for systems and services of intelligent transport II., ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

REFERENCES

- [1] Ahsan, K., Vijay, P., "Temporal Databases: Information Systems", Booktango. 2014
- [2] Ashdown, L., Kyte T., "Oracle database concepts", Oracle Press. 2015
- [3] Fekih, R., Jemili, F., "Distributed Architecture of an Intrusion Detection System Based on Cloud Computing and Big Data Techniques". In 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunication, SETIT 2018; Hammamet; Tunisia, 2018
- [4] Fernandez, J., Romera, J., "Indexes to Find the Optimal Number of Clusters in a Hierarchical Clustering". In 14th International Conference on Soft Computing Models in Industrial and Environmental Applications, SOCO 2019; Seville; Spain, 2020
- [5] Hendawitharana, S., Priyasad, M., Rajapakse, R., "Comparative Study of Spatial and Temporal Variation of Drought Using Remotely Sensed Data". I, Lecture Notes in Civil Engineering Volume 44, 2020
- [6] Johnston, T., "Bi-temporal data – Theory and Practice", Morgan Kaufmann, 2014.
- [7] Johnston, T., Weis, R., "Managing Time in Relational Databases", Morgan Kaufmann. 2010
- [8] Kvet, M., Matiaško, K., "Temporal Data Group Management", IEEE conference IDT, 2017
- [9] Kvet, M., Matiaško, K., "Temporal flower index eliminating impact of high water mark". In Innovations for community services : proceedings. Cham: Springer International Publishing AG, 2018
- [10] Kvet, M., Matiaško, K., "Transaction Management in Temporal System", IEEE conference CISTI, 2014
- [11] Kvet, M., Matiaško, K., "Uni-temporal modelling extension at the object vs. attribute level", IEEE conference UKSim, 2014
- [12] Kuhn, D., Alapati, S., Padfield, B., "Expert Oracle Indexing Access Paths", Apress. 2016
- [13] Kumar, N., "Efficient data deduplication for big data storage systems". In Advances in Intelligent Systems and Computing, 714, 2019
- [14] Li, S., Qin, Z., Song, H., "A Temporal-Spatial Method for Group Detection, Locating and Tracking", In IEEE Access, 4. 2016

Mind Map and Business Process Model: Specification support by model transformation

1st Ján Lang

Faculty of Informatics and Information Technologies
Slovak university of technology in Bratislava
Bratislava, Slovakia
jan.lang@stuba.sk

2nd Radovan Kohút

Faculty of Informatics and Information Technologies
Slovak university of technology in Bratislava
Bratislava, Slovakia
xkohut@stuba.sk

Abstract—One of the main reasons for the software project setback is the poor level of requirements specification. Communication between customer and developer is often a problematic part of the requirements specification phase. This also arises from the fact developers use modeling techniques specific for the software engineering domain to record requirements. However, customers often have difficulty understanding what the models are representing because they do not know the notations used in the domain of software engineering. The solution could be in using simpler, natural and comprehensible modeling techniques to most people. Some models, because of their simplicity, may be less expressive than the more complex models. But they could be able to help customers to understand what the software engineering models represent. The paper deals with the transformation between these two types of models.

Index Terms—Mind map, business process model, transformation, requirements, specification

I. INTRODUCTION

Proper specification of requirements is one of the key steps in software development. Without good defined requirements specifications, the developer does not know exactly what to create and the user does not know what to expect from the software. Many times the developer does not know what software the customer needs and incorrectly defines the requirements. It is also not possible to verify that the software created meets the customer's expectations when it is insufficiently defined. Communication between the customer and the software developer is very important in creating the requirements specification. During this communication, the developer records customer's requests for future software while trying to understand the domain. However, what the developer learns from the customer must also be recorded in some way. Very often the voice-recording is being made within the interview with the customer. Such a capture can serve to further analyze the interview conducted to refine and supplement other software requirements. But it is not too much a convenient way in defining requirements. Each view of the defined requirements would require the need to listen to the recording several times. This method would be impractical and time-consuming. Therefore, software engineering uses applications to capture requirements e.g. graphically. Developers often use graphical rendering to define requirements by notation from a software engineering domain such as Unified

Modeling Language (UML)¹ or Business Process Model and Notation (BPMN)². The problem is that the customer may not understand these notations. Studying these graphical notations would be time-consuming for the customer and a little bit difficult as well. Besides, to the graphical notation used in the software engineering domain, there are notations that are easy to understand and learn. One such notation is the Mind Map (MM). Modeling through MM is simple and intuitive. However, the notation of the MM is very brief compared to the notations used in the software engineering models. One solution would be a mutual enrichment of these two models. However, there is a need to resolve the transfer between the individual notations. Given the difference between the two notations, the transformation will not be trivial. The vision is to allow the customer to express and model the requirements in a simple notation that the customer understands – using MM. The assumption is that this simple model can support a more complex Business Process Model (BPM) in BPMN used by software and business engineer. The challenge here is the transformation of these models. The question is the possibility of the realization, usefulness of the result, etc. The transformation from the more complex BPM model to a simple MM model is also questionable. The paper explores the possibilities of the Mind Map to the BPMN transformation and vice versa. For this purpose, a transformation tool and extension of the existing Mind Maps editing application (Xmind³) were implemented. Section 2 presents the domain-specific ideas of communication issues and related work. Section 3 and 4 depict BPMN and MM as a means of expression. Section 5 presents the proposed approach. In section 6 an evaluation is given and finally, in section 7, there is a conclusion.

II. DOMAIN-SPECIFIC IDEAS COMMUNICATION ISSUE, RELATED WORK

The specification describes and defines the various requirements that the proposed software must meet [2]. There are several classifications of requirements. But the functional and non - functional requirements are known. These requirements should comprehensively describe the purpose of the software.

¹<https://www.omg.org/spec/UML/>

²<https://www.omg.org/spec/BPMN/>

³<https://www.xmind.net/>

It is very important as it can save time and effort for software developers. There are various techniques for obtaining and defining requirements [4]. However, each technique is based on customer-developer communication. In practice, inadequate communication proves to be one of the most common problems in identifying and defining software requirements. Several studies [5], [7], [10] have been carried out describing various types of software system design problems. Most of them, if not all, are the result of communication failure. This failure can cause the software being created to work incorrectly, contain many errors, or even cause the entire project to fail. That is why more attention needs to be paid to the early stages of the requirements specification [3].

The transformation between models is a well-known topic in software engineering. However, in the context of BPMN, more attention is paid to UML-bound transformations [1], [6]–[8]. The initial requirements specification phases can be done informally without any supporting tool. However, if the complexity of the domain for which the system is created is large, it is obvious that specification and definition of requirements will not be possible without a suitable tool. The problem, however, is that the customer often does not know, does not use such tools and even does not understand the resulting formal models. The specification approach by use cases (Use Case Driven approach) is significantly used to model requirements, nowadays [9]. The result is made up of two parts - a use case diagram and a detailed description of the use cases, which is of much greater importance for its implementation. Use case diagram respects the prescribed UML notation and only provides an overview of use cases. However, the UML standard does not specify its descriptive form even though there are multiple approaches e.g. Jacobson's, Cockburn's, Palmquist's and Overgaard's, etc. Several non-trivial tools implementing UML standards are also available (Enterprise Architect, Rational Software Architect, etc.). Ultimately, the domain expert carries the idea that needs to be postponed to the developer. But the process of exchanging information between stakeholders is not trivial. The current state of the art distinguishes three main communication barriers: the inefficiency of the current communication channels, the limited ability of expression due to notations and social and organizational barriers [12].

III. BUSINESS PROCESS MODEL AS A TOOL FOR DEFINING REQUIREMENTS

Business process modeling can be used to define and specify requirements. Formalization of this kind follows its standard in the form of a BPMN under OMG. It has a relatively rich notation. This approach is mainly used in the early stages of software projects. Fig. 5 at its top part shows an example of a business process model that captures the process of delivery shipment from submission to delivery to the recipient. Tasks are shown as rounded rectangles and each gate is diamond-shaped. The flow between components is indicated by oriented arrows.

IV. MIND MAPS AS A TOOL FOR CAPTURING PROCESS PROPERTIES

Mind maps can be used as a simple modeling tool for a user to capture the details of the processes running in a given domain. A mind map is a diagram for representing tasks, words, concepts, or notes attached to a central concept. Creating a mind map is very easy. In the center of the map is a central concept to which other related terms are connected. Because mind mapping modeling is simple and intuitive, the customer could view the properties of a certain process in their domain without having to know the complex notations used by software engineering diagrams. Fig. 5 in its middle part shows an example mind map that illustrates the properties of the delivery shipment process. We can notice the differentiation of individual terms of the map. Terms depicted in colors are related to each other and belong to a common group or hierarchy of terms. There are several tools for modeling or visualizing MM. After analyzing especially those Open Source applications, Xmind was chosen mainly because of better community support and availability on GitHub⁴. From the foregoing, it is considered to easily apply and support the specification through MM both for developers and domain experts as well.

V. MODEL TRANSFORMATION: PROPOSED APPROACH

This section provides insight into the applied approach of model transformation.

A. MM to BPM transformation

If a customer has to model the processes in related domain using a MM, the knowledge that the mind map captures, it can be assumed, it should enrich another more complex model. Since MM is a very simple model, it would not be sufficient to define the requirements precisely neither to enrich more complex model. However, MM has the advantage of requiring almost no knowledge. So the customer can use it to model related domain. In this way may be also possible to obtain knowledge and domain information from the customer.

The mind map captures the main concepts of the domain and also connects the individual concepts that are related. It also creates a hierarchy of these terms based on how they relate to each other. These terms are compared with those captured in the business process model. If there are no connections in the business process diagram, the tool will recommend them to the diagram. An important feature of the business process model utilized in the transformation is that the whole process takes place in one main swim-lane. The name or description of the swim-line corresponds to the name of the business process. We assume the central concept of the MM corresponds to the main topic to which the mind map relates. Similarly, the business process model name approximates what process is modeled by the given diagram and thus basically also approximates the main idea of this process. The task names in the business process model are a combination of a

⁴<https://github.com/>

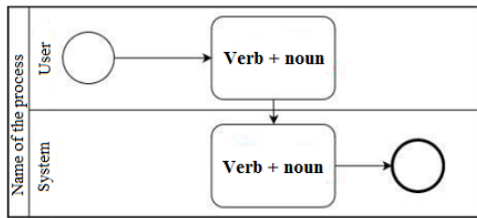


Figure 1. Assumed composition of BPMN diagram elements description

verb and a noun as another feature used in transformations see Fig. 1. This combination is just a pre-condition for our method based on the standard related statements [11]. The convention of naming a task in models by combining a noun and a verb emphasizes the achievement of a specific goal after the task. A frequent shortcoming of naming tasks in BPM is task naming exclusively by a noun. Such naming is not very appropriate. It is difficult to distinguish a given task type element from other types such as an event, data object or process area, based only on their name. Since, mainly verbal description for mapping of individual elements of one model to another will be used, observing such a description of tasks in the model is practically necessary for the correct functionality of the created transformation tool. Similarly, the use of long task names business process model is not appropriate. Such an approach, instead of emphasizing the goal of the task, emphasizes the description of the task thus directing the diagram rather than a text document. A more appropriate approach is to use a short description of the task, highlighting the objective, and then describing the task in more detail in the documentation or comments. Fig. 1 shows BPM elements – the swim-line, user-roles and the tasks in the business process model. It also states tasks in the form: verb + noun. This is important for the model transformation tool to work properly. Fig. 1 is defined to give a better idea of the placement of the task description. Subsequently, tasks descriptions are MM concepts’ derivatives.

B. BPM to MM transformation

Implementation of this part of the prototype tool has been realized in Java concerning the available DOM (Document Object Model) API and SAX (Simple API for XML) API. Both models are maintained as XML documents. When transforming from a business process model to a mind map, task descriptions are mapped to concepts in the mind map. The central concept of the mind map is determined by the description of the main swim-line representing the process. This description also corresponds to the name of the business process. When transforming from BPMN to MM, it is movement from a model with a richer informative value to a simpler model with a limited informative value. It is, therefore, possible that some information and the connections between them will be lost during the transformation as well.

The algorithm of this transformation is shown in Fig. 2. First, an XML DOM document is created from the specified input file. Furthermore, the name of the main swim-line is obtained, which will form the name of the main concept of

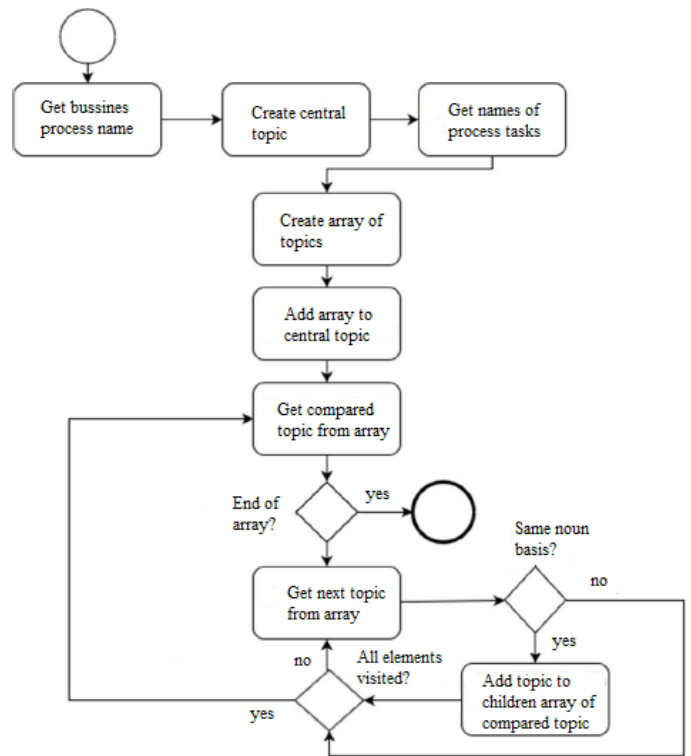


Figure 2. Algorithm of concept processing from BPMN diagram.

the future MM. Subsequently, all other terms found in the business process model are obtained. These are stored in a structure represented by an array of terms. The assumption is that the descriptions of individual tasks in the business process model consist of a verb and a noun composition. The model with the tasks described in this way meets some recommended convention. Such a convention, when entering task descriptions, forces the person who creates the model to focus exactly on what is happening during the task.

The last step is to go through the field of acquired terms and find out the links between them. The links between the concepts are determined by comparing them based on their names. If a term with the same word base is found, it is then added to the term field of that term that matches. If the word bases of the two terms match, they are then connected and displayed in the same color on the map. Visualization in the same color is a guide for the user. In this way, it can easily exchange positions of individual-related terms. Thus, it may vary how related terms are interconnected. The word base is derived from a noun using the word base tool (stemming). If the word bases of the two terms match, then a term with the same word base is assigned to the field of the appended term. If the task description in the models does not contain a noun, it is automatically assigned to the central concept of the mind map. Due to the difficulty of implementing a brand new graphics editor that would display the generated mind map, the existing tool was used. The selection was preceded by an analysis of the Open Source mind map editors.

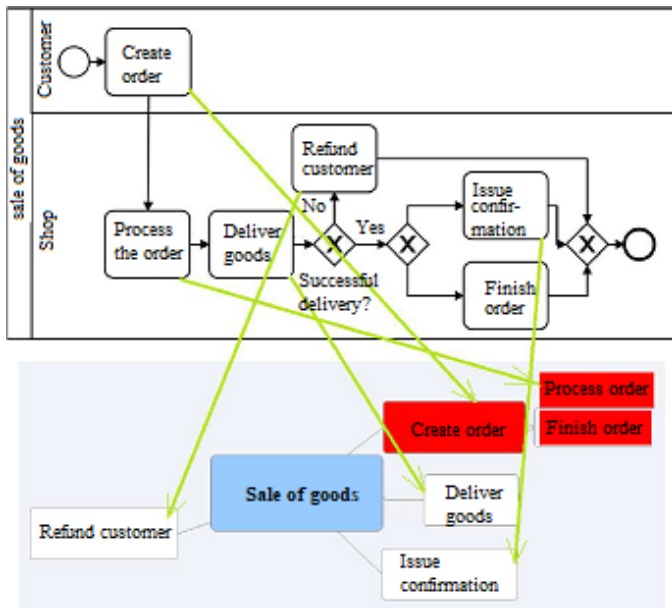


Figure 3. BPMN into MM diagram transformation example.

VI. EVALUATION

In this section, the functionality testing of the created tool is described. Individual test scenarios are designed to demonstrate the ability of the tool to transform between mind map models and business process models.

A. Transformation of BPMN diagram to MM

To realize this scenario, 5 business process-models were transformed into mind maps. The main parameter of observation was the ratio of the number of tasks in the business process model to the number of corresponding terms in the mind map. We have also observed whether the tool correctly determines the main concept of the mind map, and also whether the individual terms were placed in the right hierarchies on the basis that they were related. In Fig. 3 can be seen that the tool correctly identified the main central concept of the mind map from the business process model. We can also state that all tasks of the business process model are mapped to concepts in the mind map as well.

B. Transformation of MM to BPMN diagram

The purpose of this testing was to demonstrate the ability of the tool to transform the mind map into a business process model. The assumption was that the tool would be able to incorporate all concepts from the mind map into the business process diagram. However, moved from a simpler model to a more complex one, it is expected to lose individual connections between the tasks displayed by the business flow chart. We could not get information from the mind map in which the order of the individual tasks of the diagram should be linked. We also did not have information about the individual flow of the objects. However, it remained important that the tool offered at least a list of all tasks that should be in the diagram.

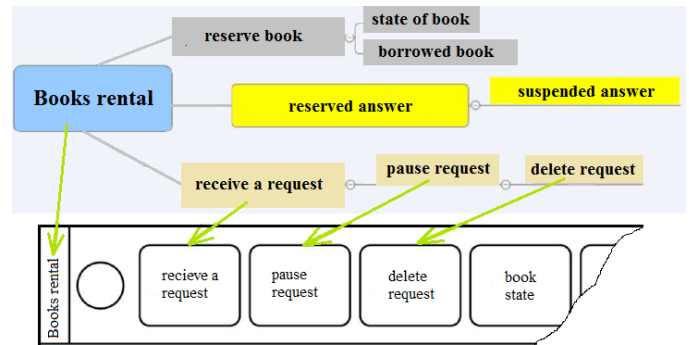


Figure 4. MM into BPMN diagram transformation example.

It is then up to the user as the individual elements of the tasks to connect, so the diagram correctly describes the process of the modeled domain. Such a list of tasks that the diagram should contain can make a significant contribution to modeling the customer's domain by the developer. Just imagine that a developer is modeling a customer domain that the developer does not know too much or even at all, the customer will provide him a mind map describing the domain. Because the customer knows their domain and processes in it reliably, the customer can create a mind map that contains all the important terms from that domain. There is a good chance that such a map will also contain terms that a developer who is not an expert in the domain would not think of. Consequently, transforming such a mind map created by the customer into a business process model can enrich the model with new features that would otherwise go unnoticed. In Fig. 4 is shown a mind map that describes the process of lending books. This example transformation is intended to demonstrate the conversion of the mind map into a business process model. Individual mappings are indicated by green lines connecting map concepts to task elements in the BPMN diagram. We can see that after the transformation, the diagram contains all the terms of the map. These are displayed as individual tasks in the diagram. They are arranged according to how they were included in the individual hierarchies of context in the mind map. The hierarchies are distinguished in the mind map by colors. For example, it can be seen that the elements belonging to the gray hierarchy are stored in the diagram immediately after each other. The tool was able to incorporate all the concepts from the mind map into the business process model. It also correctly identified the name of the main swim-line in the diagram. The tasks in the diagram are displayed in the order corresponding to the inclusion of the individual mind map concepts in the hierarchies.

C. Transformation from BPMN to MM and back to BPMN

The following test scenario is intended to test how the business process diagram changes if it is transformed into a MM and then back to the original diagram type. The assumption is that the connection between the individual tasks of the diagram will loose. However, it is expected that the resulting diagram, after performing the two transformations,

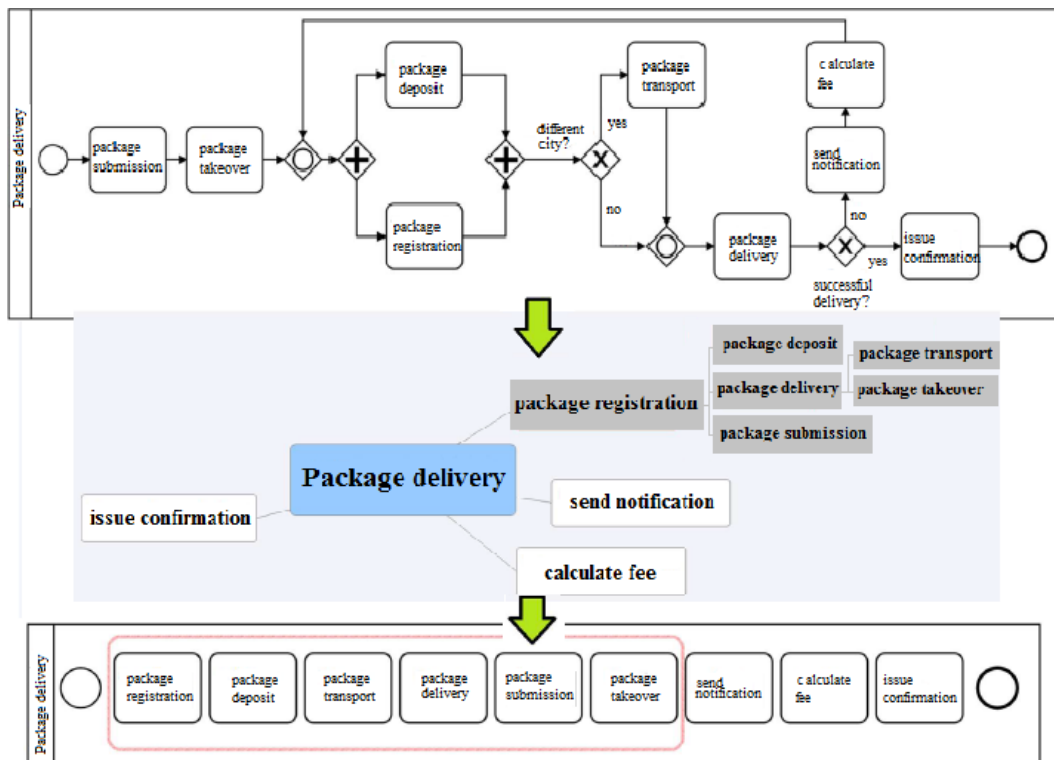


Figure 5. Multiple transformation BPMN to MM to BPMN.

will still be able to list all the tasks contained in the original diagram. Assuming the resulting model see Fig. 5 contains an ordered list of tasks, which also contained the original diagram. However, the resulting diagram no longer contains diagram objects such as gateways or connection objects that would link individual tasks. During the double transformation, none of the tasks of the original diagram were lost. Only the connections between tasks were lost. Each of the diagrams contains an equal number of elements corresponding to the concepts in the mind map and tasks in the business process diagram. We may also notice the inclusion of a group of ideas in the mind map in its hierarchy marked in a red rounded square. The tool has placed these elements in a separate hierarchy because they are all related to the shipment. Uniform color marking is helpful if the user wants to rearrange the terms on the map. By color, the user knows which terms could be moved within their hierarchy.

D. Transformation from MM to BPMN and back to MM

This scenario is designed to test how the mind map changes during the transformation into a business process model and then back to the original mind map model. Mind Map is a simpler model that does not provide, for example, objects such as gates or flow objects the business process model provides. At the same time, the individual terms in the map are linked to each other according to their context, which is based on the meaning of their description. Therefore, it is assumed that transforming it into a business process model no connection between the elements will be lost. In Fig. 6 it can be seen

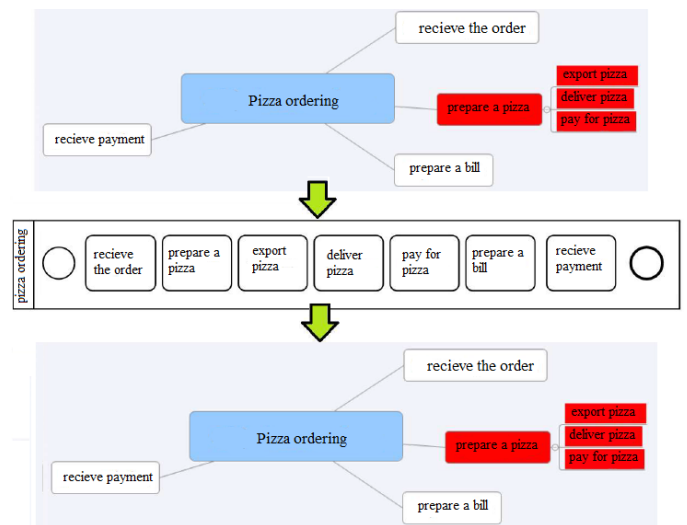


Figure 6. Multiple transformation MM to BPMN to MM.

that the result after the second transformation corresponds to expectations and the mind map coincides with the original one. So, the relationship between the concepts found in the original map will not be lost during the transformation into a richer model.

E. Existing BPMN to BPMN obtained from MM comparison

The following test scenario was intended to test the benefits of using a transformation tool to transform a mind map to

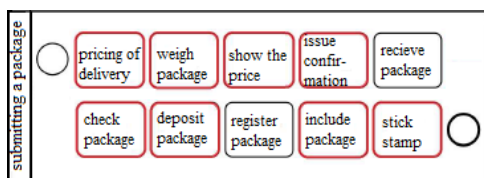


Figure 7. Overview of BPMN diagram tasks created from mind map

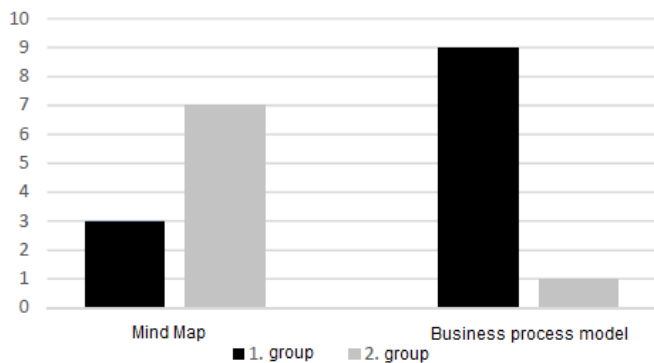


Figure 8. Modeling technique preference

obtain individual business process model tasks that could complement any business process diagrams already created. Such use of the tool could find its purpose, for example, in situations where a developer is trying to capture a customer domain which the developer has not known before. The business process model transformed from a mind map to compare with the model already available to the software developer is shown in Fig. 7. In red are the elements of the diagram tasks that were not captured in the already existing diagram. Only two tasks are shown in a black match. The red tasks thus represent an incentive to extend the existing BPMN model.

F. Modeling technique preference

The test was aimed at determining the preferred modeling technique. The testing was carried out in the form of a questionnaire in which 10 participants participated. The first group of five participants consisted of computer science students who have already had experience with BPMN. The second group of five participants consisted of students who had not yet had experience with BPMN. Both groups were given the task of recording a process in any domain by either BPMN or MM access. They were then asked a question: *Which modeling technique would you choose to capture processes running in any domain you elect?* In Fig. 8 can be seen mind map and business process model preferences statistics. Our assumption, people who have not yet encountered the business process model would prefer modeling by the mind map was correct.

VII. CONCLUSIONS

This work aimed to support communication in the early stages of software development between a domain expert and a developer by transforming specifications between a mind

map model and a business process model. Analysis of the domain-specific communication between the developer and the domain expert was realized. Subsequently, BPMN and MM approaches were described. The paper offers a description of the proposed approach of two-way transformation between models. As a result, transformation is possible. However, in the case of transformation from a model with a richer informative value to a simpler model with limited informative value, some information is being lost. A positive finding is that the transformation of MM into BPMN can be useful and can provide the necessary information to carry out the early stages of software development. The work also results in a tool that can transform the mind map and business process model, as well as extending the existing Xmind tool. The survey shows that inexperienced BPMN users like to use MM modeler as a tool to express the characteristics of the business process being monitored. By continuing this work, it is possible to implement additional transformation algorithms into other models specific to the software engineering domain.

ACKNOWLEDGMENT

The work reported here was supported by the Scientific Grant Agency of Slovak Republic (VEGA) under the grant No. VG 1/0759/19 and by the Research & Development Operational Programme Research and Innovation for the project Research of advanced methods of intelligent information processing, ITMS NFP313010T570, co-funded by the European Regional Development Fund.

REFERENCES

- [1] AL-FEDAGHI, S., AND ALDUWAISSAN, Y. Modeling of an enterprise and information system. In *Proceedings of the International Conference on Geoinformatics and Data Analysis - ICGDA '18* (2018), ACM Press.
- [2] BROWN, W. H., MALVEAU, R. C., MCCORMICK, H. W. S., AND MOWBRAY, T. J. *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*, 1st ed. John Wiley & Sons, Inc., New York, NY, USA, 1998.
- [3] CHARETTE, R. N. Why software fails [software failure]. *IEEE Spectrum* 42, 9 (Sep. 2005), 42–49.
- [4] COUGHLAN, J., AND MACREDIE, R. D. Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering* 7, 2 (June 2002), 47–60.
- [5] JIANG, J., AND KLEIN, G. Software development risks to project effectiveness. *Journal of Systems and Software* 52, 1 (2000), 3 – 10.
- [6] LÜBKE, D., AND SCHNEIDER, K. Visualizing use case sets as BPMN processes. In *2008 Requirements Engineering Visualization* (Sept. 2008), IEEE.
- [7] PAECH, B., DORR, J., AND KOEHLER, M. Improving requirements engineering communication in multiproject environments. *IEEE Software* 22, 1 (Jan 2005), 40–47.
- [8] PARK, G., FELLIR, F., HONG, J.-E., GARRIDO, J. L., NOGUERA, M., AND CHUNG, L. Deriving use cases from business processes. In *Proceedings of the Symposium on Applied Computing - SAC '17* (2017), ACM Press.
- [9] PROBASCO, L., AND LEFFINGWELL, D. Combining software requirements specifications with use-case modeling.
- [10] ŠTOLFA, S., AND VONDRÁK, I. A description of business process modeling as a tool for definition of requirements specification.
- [11] WEISS, P. Modeling of service-oriented architecture: Integration of business process and service modeling. 79–92.
- [12] YU, E. S. K. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering* (Jan 1997), pp. 226–235.

Comparative study of relational and graph databases

1st Malgorzata Lazarska

Czestochowa University of Technology
Institute of Computer and Information Sciences
 Czestochowa, Poland
 gosialazarska@gmail.com

2nd Olga Siedlecka-Lamch

Czestochowa University of Technology
Institute of Computer and Information Sciences
 Czestochowa, Poland
 olga.siedlecka@icis.pcz.pl

Abstract—We live in a time of incredible technological development. In a fraction of a second, millions of information are generated and processed all over the world. The need to develop new models of databases is related to the dynamic and the amount of data stored. Using database schemes which are rigid and do not quickly adapt to changes occurring in the real world is ineffective and sometimes simply impossible. The solution of all these problems could be, among others, graph databases.

In this talk, we present a detailed comparison of relational and graph databases, compiling the stages of design, implementation and testing. At each stage, we keep the same data in both databases, checking how easily they can be modelled in each logical model, then imported, and finally made queries and modifications on them.

Index Terms—graph databases, database modelling, databases optimization

I. INTRODUCTION

The modern world is in the phase of the technological revolution in which information is the strongest currency. Information in the form of data must be classified, ordered and stored. New professions related solely to data support have been appearing for several years. Separate teams deal with data analysis; separate teams deal with database design, yet another implementation, and subsequent support. Since such a staff of people is associated with particular stages of data life, how complex is this task?

In the 70s, the first drastic change in the approach to databases was noted - relational databases appeared [3]. Their model, intuitive and firmly embedded in mathematics, has worked perfectly on the market for decades [7, 17]. The ease of designing data in the form of two-dimensional tables, the ease of searching for connections, and the perfect integrity constraints are some of the many advantages of relational databases [4, 5]. SQL is one of the few, if not the only language that has survived almost 50 years without significant changes in its main structure. So what happened that the IT market began to look for other database solutions?

In addition to the undeniable advantages, the relational model also has disadvantages [12, 14]. These flaws were not severe until the world was navigating with such a big amount of data. Over the past few years, the world has produced more data than since the beginning of humanity. These data are often

difficult to insert into rigid structures; data are dynamic and unpredictable [15].

One of the possible answers to existing problems with databases are graph databases [13]. As intuitive as relational databases, equally well embedded in mathematical theory, this time in graph theory. These databases can be flexible and dynamically adapt their structure (and not only data) to changes taking place in the real world. The first commercial systems implementing graph databases were created in the early 20th century. In recent years, they have been gaining popularity, at least due to the strong position of such systems as OrientDB or Neo4j [6, 19].

Among the ongoing research in this area, there is no lack of work on the efficiency and benefits of existing graph databases. Scientists are looking for optimization and security solutions for them [9, 11]. However, there are no detailed comparison and compilation of all stages of the database life in two such extremely different models as relational and graph models.

In our paper we present stages of designing, implementation and testing of relational and graph databases. All the stages are thoroughly compared.

II. DATABASE MODELLING

The process of designing any type of database could be divided into several stages [4, 8, 10]. The first step is to state the clients demands and the analysis of all data and sources retrieved from the customer.

On the following stage analysts, or in the majority of cases, a team of developers, moves to conceptual modelling. This stage aims to identify the data (of the most significant importance) which should be stored and updated according to the current needs. The effect of this modelling stage is usually a scheme of entity-relationships or a UML scheme and optionally the descriptions of types and relationships which were not able to include in the scheme [1, 16].

The next step is to move to the given logic model. After the conceptual stage, it is possible to estimate the level of data complexity. At this moment we can determine whether we are supposed to work on simple, not nested data, the configuration of which is predictable and so is the future database usage or on the data which are hierarchical, complex and structurally unpredictable. Depending on the assumptions mentioned above and clients financing potential, designers choose relational (or to be more precise - a post-relational

The project financed under the program of the Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in 2019 - 2022 project number 020/RID/2018/19

model), semi-structural, object, graph or another database model [18].

During the logical modelling phase, the development team has already assumed on which DBMS they will be working and has chosen the types, identifying properties and relations accordingly. They usually add specific optimization tools to the given project, for instance, indexes, clusters, or perspectives in case of the relational model. This stage is followed by the relevant project implementation and testing. We use in our research and presentation of the next project phases an artificially generated databases of phone calls made in the United States.

A. Conceptual database modeling

In the first phase of conceptual modelling, we had to emerge entities which store the information of the paramount significance. In the project of phone calls, we have distinguished the entities of Addresses, Calls, Messages, Phone_Numbers, SIM_Cards and Customers among whom we can divide entities of Corporate_Customers and Individual_Customers. Moreover, we had to define each entities properties. For example, we need the time of start and end of a call, building number or the name of a street. Among them, we will find identifying properties, descriptors, and we determine whether the properties are mandatory or optional.

The last stage is to determine the relationships which bind the given entities. For each relationship, we determine its type. Unary relationships are those binding the instances of the same entity. For instances of two entities - we have got a binary relationship. The last example and the most difficult to find - the simultaneous ternary relationships. We also define the association type for every relationship. If we bind one instance of the first entity with one instance of the other entity, there appears a 1:1 association, for example, one telephone number - one SIM card. If we bind one instance of the first entity with many instances of the other entity, there appears "one to many" associations - a 1:N, for example one client, many telephone numbers. The last association is an N: M, meaning "many to many", which by analogy binds many instances of one entity with many instances of the other entity. Lastly, we define if the relation is obligatory or optional (there may exist telephone numbers which are not bound to any client) [1, 16].

The next stage is finding the hierarchies and relationships which excludes one another. In our case, there exist clear subtypes of individual and business client.

In the figure 1 we illustrated the final entity-relationship scheme. On this stage, we should choose the type for logical modelling by evaluating all pros and cons. If the data are simple, they structure themselves into two-dimensional tables, their development is predictable, and we are sure that the designed structure will not change in time, we know that the best choice will be the relational model. However, if in our ER model appears hierarchies, N:M relations, complex data or the perspective data would develop in the unknown direction, we

should opt for a different model. In our research, we decided to focus on the graph model.

B. Relational data model

If, for example for economic reasons, we decided to choose relational model, then the main problem would be the queries optimisation or the high costs of join operations on many tables or generating many tables having of a high number of columns the value of which is in the majority of cases unknown. By moving to the relational model, the entities in the majority of cases have become relations and their properties the attributes. By natural, the identifying properties have changed into primary keys. The obligatory properties have received the limitation NOT NULL. We have specified the types by adapting them to their future implementation in DBMS Oracle.

Let us now look at the entities diagram from the previous subsection. We decided to generate superior relation Customers and two subordinate relations Individual_Customers and Corporate_Customers. We believe that inheritance is, in this case, a better solution that generating a massive relation with many unknown values. Applying subordinate relations has further implications for generating a table creating the script and the data import itself.

The next stage of relational modelling is to change entities relationships into relations relationships or other distinctive relations for N:M relationships. The model we analyse has eight tables and seven relations. The relationships between Customers and Individual_Customers, but also Customers and Corporate_Customers are mutually exclusive. The given customer can only be an individual client or corporate client. Some of the attributes are not obligatory, for instance email or Marital_status. The final model structure can be found in figure 2.

C. Graph data model

Graph model tends to be more intuitive for data of phone calls because it reflects the network of such connections in a more natural manner. To examine how does the relational and graph databases comparison look like in reality, we modelled the graph database in a very similar way as the relational database.

In the graph database, we define nodes and relationships, whereas each node and relationship can have its labels and properties. Moving from relationship-entities model to graph model, we have changed entities into labels accordingly. Entities instances have become the nodes. Entities properties have changed into nodes properties. Finally, the edges binding the nodes in the graph match the relationships in the ER model.

Still, we want our graph database to match the relational database of the given phone calls from figure 1 as much as possible. If so, each node in the graph corresponds with one tuple from the relational database, meaning a specific client, telephone number, address or other data. Each node has the same properties as call start time, text message sending time,

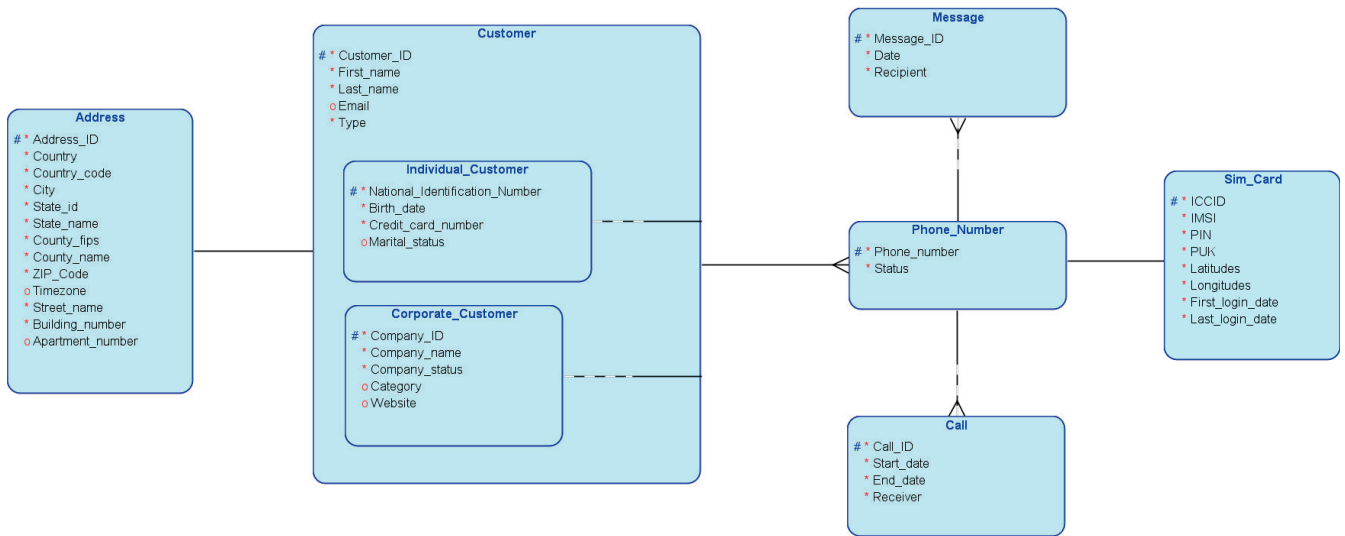


Fig. 1. Conceptual model of a relational database of phone calls made in the United States

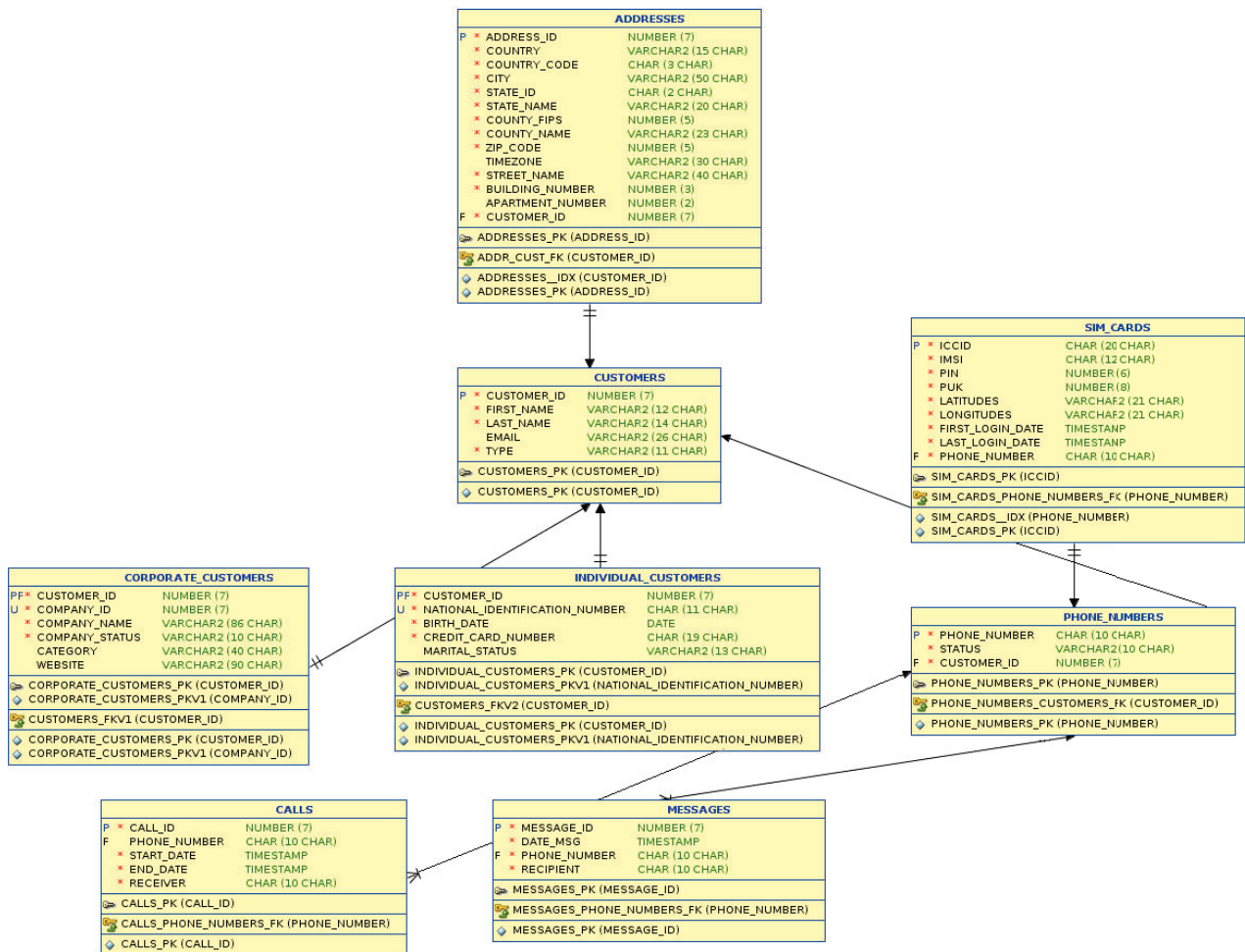


Fig. 2. Relational model of a relational database of phone calls made in the United States

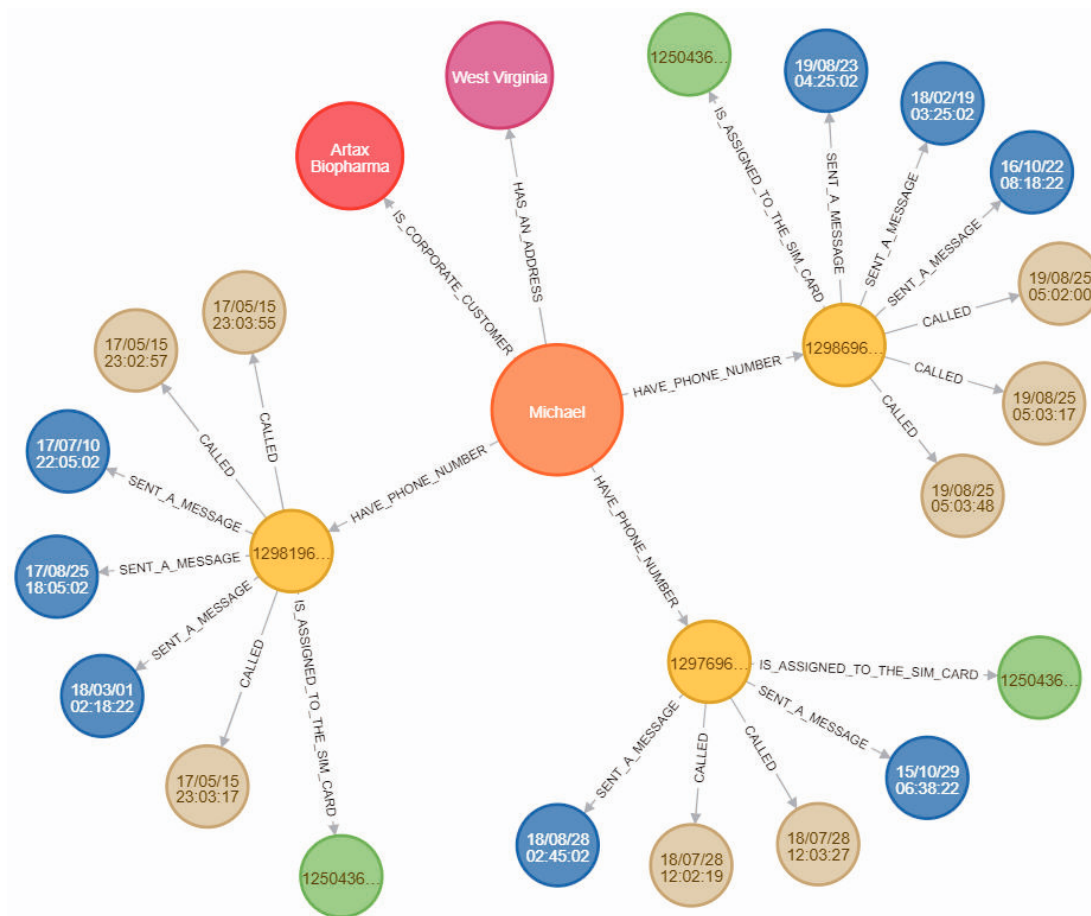


Fig. 3. Fragment of a graph database of phone calls made in the United States

property number or SIM card number as in the relational database.

In the figure 3 there is shown a fragment of graph database which reflects one specific client and all his data gathered in the database. The data we can see in each node is the value of the specific property of this node. Each relationship is described and directed. Additionally, each of the nodes mentioned above possesses the specific label, not visible here, which corresponds to the entity name and relation form relational database.

III. DATABASES IMPLEMENTATION

It is essential whether we choose relational or graph database to store our data, but first, we need to get acquainted with the given data and know more details about them. While creating the conceptual model of a database, we were aware of the type of data we wanted to store. We had already known which data are optional and which are obligatory.

Phone call data can be considered a right research subject for they can create complex structures. Each client may have millions of phone calls and send millions of text messages. Moreover, she/he does not have to limit herself/himself to one telephone number which possess the excellent potential for

analysis, but could also generate difficulties with storing such data with effective exploration of them at the same time.

A. Data generation

The first stage of work on performance comparison of relational and graph databases was to generate data. We used the RStudio environment. Having developed the conceptual data model and knowledge on what structure the relationships would have, we could generate pseudo-random data for our database using R language.

Customers table has the highest number of relationships. Five hundred thousand records of Customer_ID, First_name, Last_name, Email and Type attributes are the data prepared for the import to the Customers table. Each of the clients can be either an individual or a corporate client. Corporate clients data equals 100 000 records and 400 000 records for individual clients. Each of the clients has an assigned address. For simplification, we assumed that each of the clients had provided only the company's office, and for each of the client, there is stored only one address meaning that we also have 500 000 records.

Phone_Numbers is another set of entities which is bound by many relations. We assumed that in our database there are

1 200 000 of registered telephone numbers. The same number of the records is stored for SIM_Cards table because each of the telephone numbers has one assigned SIM card. Additionally, we have generated three millions of records about phone calls. The highest number of records belongs to the number of text messages, which equals 3 100 000. The combined number of generated data equals ten million records.

Thanks to exporting all generated data to .csv format the import to DBMS Oracle did not impose any difficulties. Unlike the Excel format, the .csv format does not have lines limitation making it more convenient and in this way the best for our needs.

B. Relational database

We employed DBMS Oracle to generate the relational database of phone calls. The first step was to design the conceptual model in SQLDeveloper Data Modeler. The created model can be found above in figure 1. On the outset, we have defined the table. Then we have entered the columns by defining their types, and whether they are obligatory or optional. At the same time, we have also defined whether the columns are primary keys.

In the following step, we defined the relationship binding the tables together. We have determined the hierarchy between the Customers, Corporate_Customers and Individual_Customers tables because we have considered it the from the performance and allocated memory perspective as the best solution. This solution increases the number of tables combined by one because the client can be only corporate or individual.

Then, we have generated tables with columns from the relational model. In the scheme we have created, the foreign keys, assigned by relations between objects, have already been automatically added in the tables. We needed to apply small corrections to our model because of the limitation of the attributes and relations names length to 30 signs. The over accumulated foreign keys, remaining in both of the tables, were also erased, for there was only one foreign key needed in one of them.

After corrections, we have generated DDL script, the launching of which has automatically created eight tables in our database. This script also contained commands that add primary and foreign keys to each table. All created keys can be seen in figure 2, marked with letters P and F. When creating the primary key on the selected column, a unique index on this column was also automatically created. We have also added constraints unique to the National_Identification_Number column in the Individual_Customers table and the Company_ID column in the Corporate_Customers table. Besides, we have created a unique index on the Customer_ID column in the Addresses table and on the Phone_number column in the SIM_Cards table.

We could start the realization of the next phase only after the tables forming, creating indexes and constraints. Having already prepared tables with indexes and constraints and generated in .csv format data, we started to import the data to the database using Oracle SQLDeveloper. We have chosen

Insert import method, so lines were imported one after one to the database. The data import was very efficient.

In the relational database, we have to focus on the order in which the data are imported to tables. Initially, we had to fill in the Customers table because the Customer_ID column is a foreign key in almost all other tables.

C. Graph database

To create the graph database, we have chosen the Neo4j system in Enterprise version. It constitutes one of the most popular and most user-friendly graph database systems. The easiest method of the data import from DBMS Oracle to Neo4j is to use the Neo4j ETL tool. We could get access to the database by making a JDBC connection. Transformation of the relational database into graph database does not require from us an extensive workload, so meanwhile, we could add corrections as set the graphic design of the graph.

However, we have chosen an alternative solution because we aimed at testing the import method similar to DBMS Oracle. We decided to directly import the data to Neo4j from the .csv files as in case of a relational database. The name of the table was corresponding to the nodes label; the tables columns were corresponding nodes properties and relationships between the tables has been moved onto relationships of the nodes. In total, we imported 10 million records, which gave us 10 million nodes.

The order of the data import was of no importance here. Each record from the file has created one independent node in the graph. After creating all nodes, we created indexes on them. If we created an index in the relational database for an attribute in the table, then we also created an index in the graph database for the adequate property of the nodes with a given label.

In Neo4j, we created indexes for the Customer_ID and Address_ID properties of nodes with the Addresses label, for the Call_ID property of the nodes with the Calls label, for the Message_ID property of the nodes with the Messages label, for the Phone_number and ICCID properties of the nodes with the SIM_Cards label, for the Customer_ID properties of the nodes with the Customers label, for properties Customer_ID and National_Identification_Number nodes with Individual_Customers label, for Customer_ID and Company_ID properties for nodes with Corporate_Customers label and for Phone_number properties for nodes with Phone_Numbers label.

We have also created some constraints, similar to a relational database. We have added constraints unique for all node properties that corresponded to primary keys in the relational database of phone calls. We have also added them for the remaining properties of nodes corresponding to constraint unique in the relational database.

The last stage was to determine the relationships between the nodes, as we had to determine in relational database the relationships between the tables. After finishing this task, we have received 9 500 000 relationships binding the specific

nodes with each other. In the figure 3 it is shown how do the relationships in Neo4j look like.

Creating the graph database and filling it in with data of the specific type was, in this case, a bit less time-consuming and required less workload than in the relational database. The significant advantage of a graph database is also the possibility to present the scores of queries in the form of a graph in the browser. Thanks to this solution, it is easier to see specific dependencies and interpret the queries scores.

IV. CONDUCTED RESEARCH

The most interesting aspect of comparison of the chosen databases is the pace of the data processing, their exploration and possibilities of their modification. We wanted to find out in what time we could draw the conclusions from the data. We also wanted to know how the operations and queries are influencing the databases efficiency.

We have conducted a number of test which aimed at providing us the answer to how costly it is for our phone calls database to perform different queries. We have taken into account the realization of the same queries in DBMS Oracle and Neo4j. In both cases they had the same data, which return the same number of the same answers to the given query.

We used SQL language to built queries for Oracle. We have deployed the queries directly from Oracle SQLDeveloper level. For Neo4j Enterprise queries we have used the dedicated to this task Cypher language. In both cases the scores were the same records, but in Neo4j we can decide in what form we want our scores to be presented. There are many occasions in which we could decide on the graphic form of the scores presentations, but it also depends on whether we choose nodes or only their properties.

A. The research dependent on the number of joined tables/labels

The very first performance test conducted by us based on relational database on a given number of tables bound by left join operation. All queries had the same fixed condition. In the graph database there were correspondingly used the same queries with the same fixed condition of searching for a nodes only with the given labels.

We have started the research from the query operating on the first table and correspondingly in the graph database in the nodes set of the same label. In both cases we have started from Customers table/label. Then we were adding the next tables/labels without changing the query's condition. We were observing in both databases the time of queries and a number of records/nodes which had to be visited to perform the query. Examples of queries can be found in table IV.

The results of the query execution time (table I) for a small number of joined tables/labels of nodes (of up to 3) are more favourable to DBMS Oracle in relational database than in Neo4j in graph database. Still, it is worth noting that when the number of joined tables in relational database grows then the scores start to grow exponentially while in the graph database

the results of the query execution time for the same conditions grows slightly and linear.

Another essential aspect is the number of visited records/nodes during query realisation. We can observe that, with the fixed condition which operates only for Customers table/label, while attaching subsequent tables/nodes of different labels the number of visited records in relational database grows rapidly and the growth in graph database is slight.

B. The research dependent on records/nodes limitation

The following performance test in the relational database was to execute all the same queries in the maximum fixed number of seven tables bound by left join operation. Only the condition which defined the number of the received records had any impact. In the graph database, we correspondingly focused on the query of the permanent structure which operated on seven nodes labels, in which we defined in the condition a number of scoring records accordingly. Examples of queries can be found in table V.

As it was shown in II, each time we were increasing the number of records tenfold. The time of queries realization in the graph database is significantly better than in the relational database. It took Neo4j only 2,2 seconds to reach a score of 100 000 records while the same operation in Oracle took 111,3 seconds.

As for the number of database reading, Neo4j increases tenfold the number of the visited objects to execute the given query accordingly to the increasing tenfold number of result records. In the relational database, although the number of records visited is greater, their growth is much weaker in the broader perspective.

C. The research on the data modification

The third test aim was to examine how costly and time-consuming is the update of the data on the tables of different sizes. Simultaneously we considered in the graph database the updates of the data marked by different labels which corresponded to tables in relational database. In all cases we were updating only one attribute/property in one record/node. In table III there are presented the outcomes of the conducted tests and table VI shows examples of queries.

As it can be observed, the update time in the graph database is in our case constant and takes for a set of nodes of any type of label exactly one millisecond. In DBMS Oracle there is a clear dependency between the size of the table and the time of its update. With the increase of the records number in table there increases also the time of the update.

As for the reading of the databases, there is also a certain difference. Each time we update the data in the relational database, the total number of the visited records equals 2. In the graph database, in this case, we need only half of such operations.

TABLE I
THE RESEARCH DEPENDENT ON THE NUMBER OF JOINED TABLES/LABELS

Number of joined tables/labels	Oracle		Neo4J	
	Query time SQL [sec]	Cardinality	Query time Cypher [sec]	Total db hits
1	0,083	2	0,180	1 001 079
2	0,085	5	0,184	1 001 083
3	0,084	8	0,181	1 001 087
4	0,257	1 200 009	0,190	1 001 098
5	0,248	1 200 012	0,187	1 001 106
6	0,568	4 200 013	0,195	1 001 141
7	0,863	7 300 017	0,203	1 001 241

TABLE II
THE RESEARCH DEPENDENT ON RECORDS/NODES LIMITATION

Number of result records	Oracle		Neo4J	
	Query time SQL [sec]	Cardinality	Query time Cypher [sec]	Total db hits
1	0,042	18	0,001	28
10	6,077	3 100 041	0,001	133
100	24,912	7 758 088	0,003	1 256
1 000	26,477	7 789 966	0,018	12 486
10 000	47,459	8 730 196	0,137	124 743
100 000	111,322	10 201 522	2,206	1 247 379

TABLE III
THE RESEARCH ON THE DATA MODIFICATION

Size of the table/set of nodes with the label	Oracle		Neo4J	
	Query time SQL [sec]	Cardinality	Query time Cypher [sec]	Total db hits
100 000	0,183	2	0,001	1
400 000	0,236	2	0,001	1
500 000	0,238	2	0,001	1
1 200 000	0,425	2	0,001	1
3 000 000	0,564	2	0,001	1
3 100 000	0,578	2	0,001	1

V. CONCLUSIONS

In this article we have presented a comparative analysis of two types of databases, relational and graph, on the example of a telephone connection database. For performance comparison, we chose the popular Oracle relational database and the Neo4j graph database. As part of the research, we went through all stages of modeling both types of databases.

The conceptual modeling stage was common. Logical modeling was simple and intuitive for both types of databases, the only obstacle was the need to maintain in the graph database the reflection of the relationship from the relational database, so that further performance tests make sense. Meanwhile, the graph model is much more flexible and enables the inclusion of complex data and dependencies.

We also showed how you can transform one model into another. It should be noted that we can transform any model of a relational database into a graph. Conversely, this is not always possible. In the graph model it is also much easier to add new properties, while in the relational database in a working environment adding new attributes can be very difficult or impossible.

In the next step, we implemented both databases. We have described each step of the procedure. The basis was the creation of a physical model of databases. Then we performed the data import.

The conducted research concerned the performance of both databases in two aspects: queries and modifications. We compared them in terms of query execution time and query cost as the number of full database reads. We conducted three types of tests to compare both databases from different sides. The obtained results clearly indicated that in our case the graph database is much more efficient in terms of time and cost of processing queries. Data modification was also faster in the graph structure.

TABLE IV
EXAMPLES OF QUERIES USED IN THE RESEARCH DEPENDENT ON THE NUMBER OF JOINED TABLES/LABELS

Oracle	Neo4j
<pre>SELECT c.Customer_ID, c.Email, i.National_Identification_Number, a.City FROM Customers c LEFT JOIN Individual_Customers i ON c.Customer_ID=i.Customer_ID LEFT JOIN Addresses a ON c.Customer_ID=a.Customer_ID WHERE c.First_name='Gertrude' AND c.Last_name='Hammes';</pre>	<pre>MATCH (c:CUSTOMERS)-[:IS_INDIVIDUAL_CUSTOMER]-> (i:INDIVIDUAL_CUSTOMERS) MATCH (c:CUSTOMERS)-[:HAS_AN_ADDRESS]-> (a:ADDRESSES) WHERE c.FIRST_NAME="Gertrude" AND c.LAST_NAME="Hammes" RETURN c.CUSTOMER_ID, c.EMAIL, i.NATIONAL_IDENTIFICATION_NUMBER, a.CITY;</pre>

TABLE V
EXAMPLES OF QUERIES USED IN THE RESEARCH DEPENDENT ON RECORDS/NODES LIMITATION

Oracle	Neo4j
<pre>SELECT c.First_name, c.Last_name, i.National_Identification_Number, a.City, p.Phone_number, s.Last_login_date, cl.Start_date, m.Date_msg FROM Customers c LEFT JOIN Individual_Customers i ON c.Customer_ID=i.Customer_ID LEFT JOIN Addresses a ON c.Customer_ID=a.Customer_ID LEFT JOIN Phone_Numbers p ON c.Customer_ID=p.Customer_ID LEFT JOIN Sim_Cards s ON p.Phone_number=s.Phone_number LEFT JOIN Calls cl ON p.Phone_number=cl.Phone_number LEFT JOIN Messages m ON p.Phone_number=m.Phone_number WHERE rownum<=100000;</pre>	<pre>MATCH (c:CUSTOMERS)-[:IS_INDIVIDUAL_CUSTOMER]-> (i:INDIVIDUAL_CUSTOMERS) MATCH (c:CUSTOMERS)-[:HAS_AN_ADDRESS]-> (a:ADDRESSES) MATCH (c:CUSTOMERS)-[:HAVE_PHONE_NUMBER]-> (p:PHONE_NUMBERS)-[:IS_ASSIGNED_TO_THE_SIM_CARD] -> (s:SIM_CARDS) MATCH (p:PHONE_NUMBERS)-[:CALLED]-> (cl:CALLS) MATCH (p:PHONE_NUMBERS)-[:SENT_A_MESSAGE]-> (m:MESSAGES) RETURN c.CUSTOMER_ID, c.EMAIL, i.NATIONAL_IDENTIFICATION_NUMBER, a.CITY, p.STATUS, s.LAST_LOGIN_DATE, cl.START_DATE, m.DATE_MSG LIMIT 100000;</pre>

TABLE VI
EXAMPLES OF QUERIES USED IN THE RESEARCH ON THE DATA MODIFICATION

Oracle	Neo4j
<pre>UPDATE Individual_Customers SET Marital_status='Married' WHERE Credit_card_number='9322-4164-9230-1468';</pre>	<pre>MATCH (i:INDIVIDUAL_CUSTOMERS {CREDIT_CARD_NUMBER: '9322-4164-9230-1468'}) SET i.MARITAL_STATUS='Married';</pre>

REFERENCES

- [1] M. Brodie, J. Mylopoulos, and J. Schmidt, On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases, and Programming Languages. Springer-Verlag, 1984
- [2] T. Clement, M. Weiyi, Principles of Database Query Processing for Advanced Applications, Morgan Kaufmann Publishers, 1998
- [3] E. F. Codd, A relational model of data for large shared data banks, Communications of the ACM, v.13 n.6, p.377-387, June 1970
- [4] T. Connolly, C. Begg, Database Systems - A Practical Approach to Design, Implementation, and Management (third edition), Addison-Wesley, 2002
- [5] C. Coronel, S. Morris, Database Systems: Design, Implementation, & Management, Cengage Learning, 2016
- [6] D. Fernandes, J. Bernardino, Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In DATA, pages 373380, 2018
- [7] H. Garcia-Molina, J. Widom, Database Systems: The Complete Book, Prentice-Hall, Englewood Cliffs, NJ, 2002
- [8] J. A. Hoffer, F. R. McFadden, Modern database management (10th ed.), Prentice Hall, 2011
- [9] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, J. Goodall, Developing an Ontology for Cyber Security Knowledge Graphs, Proceedings of the 10th Annual Cyber and Information Security Research Conference, p.1-4, April 07-09, Oak Ridge, TN, USA, 2015
- [10] D. M. Kroenke, D. J. Auer. Database Processing, Fundamentals, Design, and Implementation, Pearson (12th ed.), 2012
- [11] S.A.T. Mpinda, L.C. Ferreira, M.X. Ribeiro, M.T.P. Santos, Evaluation of Graph Databases Performance through Indexing Techniques. International Journal of Artificial Intelligence & Applications (IJAA) 6(5) 87-98 2015
- [12] L. Perkins, E. Redmond, J. R. Wilson, Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL, Pragmatic Bookshelf, 2018
- [13] I. Robinson, J. Webber, E. Eifrem, Graph Databases(2nd ed.), OReilly, Sebastopol, Canada, 2015
- [14] K. Sahatqija, J. Ajdari, X. Zenuni, B. Raufi, F. Ismaili, "Comparison between relational and NOSQL databases", IEEE Xplore, 2018
- [15] V. Sharma , M. Dave, SQL and NoSQL Databases, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 8,2012
- [16] Thalheim B.: Towards a theory of conceptual modelling. Journal of Universal Computer Science (JUCS), vol. 16 no. 20:31023137, 2010
- [17] J.D. Ullman, Principles of database and knowledge base systems, Vol. I and II, Computer Science Press, Rockville, Maryland, 1989
- [18] J. D. Ullman , H. Garcia-Molina , J. Widom, Database Systems: The Complete Book, Prentice Hall PTR, Upper Saddle River, NJ, 2001
- [19] M. Yannakakis, Graph-Theoretic Methods in Database Theory, in Proceedings of the 9th Symposium on Principles of Database Systems (PODS), pp. 230-242, ACM Press, 1990

Performance guarantees for P4 through cost analysis

1st Dániel Lukács
Faculty of Informatics
Eötvös Loránd University
 Budapest, Hungary
 dluakacs@inf.elte.hu
 ORCID: 0000-0001-9738-1134

2nd Gergely Pongrácz
Ericsson Hungary Ltd.
 Budapest, Hungary
 Gergely.Pongracz@ericsson.com
 ORCID: 0000-0002-5115-9973

3rd Máté Tejfel
Faculty of Informatics
Eötvös Loránd University
 Budapest, Hungary
3in Research Group
 Martonvásár, Hungary
 matej@inf.elte.hu
 ORCID: 0000-0001-8982-1398

Abstract—For modern switches operating with terabit-scale bandwidth it is important to achieve guarantees of high-performance as early as possible in the design and development process. P4 is the state-of-art programming language for defining the control flow of SDN network switches. The most computationally intensive part of the program flow are the lookup tables. Unfortunately, the execution semantics of P4 lookup tables is highly implementation dependent.

Continuing our previous work on P4 parsers, we propose, classify, and evaluate implementation-level cost models for P4 programs composed of parser and lookup constructs, enabling developers to estimate program execution costs (with CPU cycle precision) before actually deploying the solution to hardware.

Index Terms—P4, cost analysis, static profiling, data plane, SDN

I. INTRODUCTION

P4 [1] is a new programming language specifying packet forwarding control flow for P4-enabled switches. P4 supports the software defined networking (SDN) approach of network design. In SDN, the network is partitioned into two class of switches: data plane and control plane. Data plane switches are concerned with processing and forwarding packets (e.g. by running P4), while control plane switches (called controllers) are concerned with modifying the behaviour of the data plane switches as needed (e.g. by updating the P4 lookup tables of data plane switches).

As a networking middleware, questions regarding latency, efficiency and correctness have critical importance regarding the applicability of this language. In this work – as a continuation of our earlier work [2] –, we set out to extend our envisioned P4 cost analysis framework for handling P4 lookup tables. Ultimately, our goal is to develop a tool that can, given a text of P4 program source code and sufficient information about the execution environment (hardware, implementation, and network parameters), statically infer vital estimations of the worst case and expected costs of executing this program. The tool will aid developers and operators in calibrating both the program and the execution environment, without having to expensively deploy the system and perform blackbox tests.

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

Our selected methodology for such an analysis is incremental refinement. We see each P4 program as a high-level specification (a set of requirements) regarding the switch control flow. Unfortunately, these specifications (P4 programs) lack requirements that could help in precisely estimating execution costs (an intentional decision made by the language designers in order to aid compiler developers in viciously optimizing the language for their target hardware). To increase precision, we need to refine the specification, by adding extra information (restrictions) about the executing hardware, the language implementation, and the runtime environment. In our research, our focus is not on the data used for this refinement. Instead, our aim is to design a cost analysis tool enabling this refinement: the framework should be generic enough to support various industrially used P4 setups, and yet it should be modular enough to support effortless specialization by incrementally adding new (or better) information as it becomes available.

In this work, we discuss refinement in context of P4 match-action table lookup schemes. Section II briefly presents the technical background for understanding P4 and match-action tables. Next, in Section III, we introduce a probabilistic model we believe is generic enough for analysing all relevant P4 setups: it enables using ‘sane defaults’ in place of missing information, and at the same time supports modularly adding better quality information. Section IV and V is an exercise of refinement: we refine P4 lookup invocation using a model implementation of the DIR-24-8 algorithm, add in toy execution environment information, and validate our DIR-24-8 cost model by observing whether it conforms our expectations. Enumeration of related work and conclusion concludes this paper.

II. ABOUT THE P4 LANGUAGE

In P4 the bulk of packet forwarding work is performed by match-action tables. As a generalization of routing tables, these can be envisioned as key-value stores, where keys are patterns and values are actions. When the table is applied to a packet, the table is searched for the first entry whose pattern matches the packet, and the action of this entry is executed. P4 control flow can also depend on the executed action. Note that P4 does not define the implementation of match-action tables:

Listing 1 P4 code specifying a simple IP4 switch

```

1 #include <core.p4>
2 #include <v1model.p4>
3
4 struct headers { ethernet_t eth;
5                 ipv4_t ipv4;
6 }
7
8 header ipv4_t { bit<8> ttl;
9                bit<32> dstAddr;
10               [...]
11 }
12
13 V1Switch(p = ParserImpl(),
14         ig = ingress(),
15         dep = DeparserImpl()) main;
16
17 parser ParserImpl [...] { [...] }
18
19 control ingress(
20   inout headers h,
21   out standard_metadata_t sm) {
22
23   action on_miss() {
24
25   action nexthop(bit<9> port) {
26     sm.egress_port = port;
27     h.ipv4.ttl = h.ipv4.ttl - 8w1;
28   }
29
30   table fib {
31     actions = { on_miss; nexthop; }
32     key = { h.ipv4.dstAddr : exact; }
33     size = 131072;
34   }
35
36   table fib_lpm {
37     actions = { on_miss; nexthop; }
38     key = { h.ipv4.dstAddr : lpm; }
39     size = 16384;
40   }
41
42   apply {
43     if (h.ipv4.isValid()) {
44       switch (fib.apply().action_run) {
45         on_miss: { fib_lpm.apply(); }
46       }
47     }
48   }
49
50   control DeparserImpl [...] { [...] }

```

choosing the optimal data structures and search algorithms is the responsibility of compiler developers. Moreover, P4 programs do not describe the contents of the match-action tables: they are filled by the SDN controller during runtime. P4 programs only describe the table schema.

The example P4 code we will use is displayed by Listing 1. This program specifies a simple protocol describing basic IP4 forwarding. The elements of the forwarding pipeline is specified in the call to `V1Switch`. After headers of the incoming packet are parsed by `ParserImpl` into the structure named `headers`, the `ingress` block specifies that when the packet was successfully parsed as an IP4 packet, we apply the `fib` match-action table (exact lookup, requiring full match) to the packet. If this fails, we apply the `fib_lpm` table (LPM lookup). The `table` block `fib` declares a table schema: the `dstAddr` field of the packet is matched to table entries, which can be either one of the actions `nexthop` and `on_miss` and their arguments (e.g. the destination port). Action `nexthop` sets the `egress_port` metadata field that is read, in turn, by the switch implementation to forward the packet to the `nexthop` gateway. Then, `nexthop` decreases the IP4 time-to-live field by the 8-bit unsigned integer 1.

III. A PROBABILISTIC MODEL OF ACYCLIC PROGRAM FLOW

In the following section, we provide a probabilistic model of program execution. We then give an algorithm for calculating the expected execution cost of a program given its control flow graph (CFG). For now, we will assume that the modelled programs have no explicit loops. We exclude looping constructs from the current discussion, because (a) in the general case, calculating the expected value of a loop is undecidable, and (b) because unstructured loops (such as those constructed using jump instructions) would make analysis more involved. As we will also see in Section IV, special cases of structured loops can be easily included into the model, and this is sufficient for P4. We also omit error statements from our discussion: defining the meaning of execution cost of a program resulting in error is out of the scope of this paper. Moreover, the P4 language lacks exception handling control structures, so this omission only affects a small class of P4 programs.

A. The model

In the following, we informally assign to the control flow graph (CFG) a semantics similar to the semantics of Bayesian networks. During this discussion, we will use the CFG in Figure 1 as an example. Nodes of the CFG, called blocks (denoted as n) correspond to a value of the random variable called *program counter*: each of these values have associated execution costs. We treat CFGs as hierarchical, so we allow blocks to be mapped to lower level CFGs. A directed edge e_i between two blocks corresponds to the possible event (also denoted as e_i) of updating the program counter. Conventionally, we denote the probability of this event as $P(e_i)$. A finite *program execution* (denoted by π) is any directed path (i.e. sequence of events) selected from the CFG starting at the entry point and terminating in the exit point. Each prefix of a program execution has an associated program state. Each e_i edge has a special condition label $cond(e_i)$. If a node has multiple outgoing edges, the control always chooses the edge with the condition satisfied in the current program state. As a consequence: $P(e_i) = P(cond(e_i))$. As such, we use $P(e_i)$ and $P(cond(e_i))$ interchangeably in this paper.

We define the expected value of a g CFG as the sum of the execution costs of each execution paths, weighted by the probability of that execution:

$$E(g) = \sum_{\pi \in paths(g)} P(\pi) cost(\pi) \quad (1)$$

Since conditions depend on the program state, not all conditions are independent from each other. In other words, the probability of a condition being satisfied is a conditional probability. By the definition of conditional probability, the probability of a length-2 execution path e_1, e_2 can be calculated using the individual probabilities of each constituent elements: $P(e_2 \cap e_1) = P(e_2|e_1)P(e_1)$. For paths longer than that, we can decompose the probability of a path using the chain rule of conditional probability:

$$\begin{aligned}
 P(e_n, e_{n-1}, \dots, e_2, e_1) &= P(e_n|e_{n-1}, \dots, e_2, e_1) \\
 &\quad \cdot P(e_{n-1}|e_{n-2}, \dots, e_2, e_1) \\
 &\quad \dots \\
 &\quad \cdot P(e_2|e_1) \\
 &\quad \cdot P(e_1)
 \end{aligned} \quad (2)$$

When we talk about the cost of some $\pi = e_1, e_2, \dots, e_k$ path, we mean the sum cost of the blocks on that path. As before, we need to take into account that the execution cost of block n_i may be dependent on the path it is on:

$$\text{cost}(\pi) = \sum_{n_i \in \text{nodes}(\pi)} \text{cost}(n_i | \pi) \quad (3)$$

In most cases, we expect expressions in the form of $\text{cost}(n | \pi)$ to be directly translated to known constants (elementary operational costs), or to conditional expectations in the form of $E(n | \pi)$.

Conditional expectation provides us compositionality, which is important for efficiently computing expectation. Let g be CFG, and n be a component (a block which is mapped to a sub-CFG) of g . Let us consider random variables $X : \Omega \rightarrow \text{paths}(g)$ and $Y : \Omega \rightarrow \text{paths}(n) \cup \{\varepsilon\}$. Assuming n is deterministic, each π_y value of Y corresponds to a class of inputs of n . Inputs are generated by values of X , and so $\text{rng}(Y)$ determines a partitioning of $\text{rng}(X)$.

We should also notice that the component has no cost on those paths that do not execute it: $E(n | Y = \varepsilon) = 0$. Moreover, path π_y is only executed on π_x if and only if π_x generates an input inducing π_y :

$$P(Y = \pi_y | X = \pi_x) = \begin{cases} 1 & , \text{ if } \pi_x \text{ induces } \pi_y \\ 0 & , \text{ otherwise} \end{cases}$$

Then, by the law of total expectation,

$$\begin{aligned} E(n | X = \pi_x) &= \\ E(E(n | Y = \pi_y) | X = \pi_x) &= \\ \sum_{\pi_y \in \text{rng}(Y)} E(n | Y = \pi_y) P(Y = \pi_y | X = \pi_x) &= \\ \sum_{\pi_y \in \text{paths}(n)} E(n | Y = \pi_y) & \end{aligned}$$

This means, we can calculate $E(n | \pi_y)$ independent of π_x , and use these to calculate $E(n | \pi_x)$. The law in the form of $E(n) = E(E(n | \pi_y))$ also justifies our handling of the top-level component in Equation 1. Simple examples (assuming independence of component cost and preceding execution) of utilizing such expressions will be demonstrated in Section IV.

B. Algorithms for computing expected value

Algorithm 1 enumerates all paths and computes the probabilities and costs for each. The products and sums resulting from executing the algorithm on Figure 1 are displayed by Table I. Unfortunately, we cannot avoid enumerating all (exponentially many) execution paths induced by the CFG: if two paths merge (e.g. as in an Y-shaped component), we still need to record the different histories of the suffix in the probability conditions. On the other hand, we could take advantage of the fact that paths share some prefixes with each other in order to avoid redundantly computing probabilities and cost sums for the same prefixes. That is, given the weighted costs $(p_i p_{i+1} \dots p_n P_1) c_1$ and $(p_i p_{i+1} \dots p_n P_2) c_2$ of two paths, we can calculate the expected value as $(p_i p_{i+1} \dots p_n)(P_1 c_1 + P_2 c_2)$.

The algorithm can be optimized further by e.g. pruning low-probability branches, and selectively reversing the direction of traversal (starting from the exit).

Algorithm 1: A revisiting BFS, calculating the expected value of an acyclic CFG G from source node r

```

Input:  $G$  // an acyclic CFG
Input:  $r$  // starting node for traversing the CFG
Output: A list containing for each path in the CFG a  $(n, \pi, p, s)$  tuple
characterizing the path, where


- $n$  is the last block on the path
- $\pi$  is the path itself, i.e. a sequence of CFG edges
- $p$  is the probability of the path
- $s$  is the cost of the path


1 Function  $ExpectedValue_G(r)$ :
2    $out := \emptyset$ 
3    $q := [(r, [], 1, 0)]$ 
4   while  $q \neq []$  do
5      $(n, \pi, p, s) := Dequeue_q()$ 
6     foreach  $e$  in  $OutEdges_G(n)$  do
7        $m := Dst_G(e)$ 
8        $data := (m, [e] ++ \pi, p \cdot P(e | \pi), s + c(m))$ 
9       if  $OutEdges_G(n) = \emptyset$  then
10         $out := out \cup \{data\}$ 
11       else
12         $Enqueue_q(data)$ 
13       end
14     end
15   end
16   return  $out$ 

```

C. Data requirements

In P4, implementation of certain language elements is intentionally left undefined by the specification, so the implementors can maximize efficiency on very different executing hardware targets. Moreover, some input data is naturally only available at runtime (e.g. packets, and match-action table contents). This missing information is required to calculate the elementary costs and conditional probabilities in the expected value formula of Equation 1. But this is also the exact reason we chose this approach: new information regarding costs and probabilities can be easily added by the time and in the quality it becomes available. Until then, we can utilize sane, but far less precise defaults: we may treat all conditionals independent from each other (thus taking advantage of the fact that $A \cap B = \emptyset \implies P(A|B) = P(A)$).

This is consistent with the meaning of independence stating that any value of each variable contains no information regarding values of the other variable. We can also assign mathematical (e.g. uniform) probability distribution for each conditionals. Or we may decide to calculate conditional probabilities using static analysis techniques, or infer them with data mining on existing usage data.

We can also refine the abstraction level of the CFG (and achieve more precise results) by expanding a node into the CFG of its implementation (or a selected abstraction of its implementation). An example of this operation can be found in Section V, where we calculate the (undefined) P4 lookup costs and result probabilities by assuming that the target is using a specific algorithm (described in Section IV) to implement lookups.

IV. COST MODELS OF LOOKUP ALGORITHMS

In this paper, we model DIR-24-8, the longest prefix match (LPM) algorithm used by DPDK [3], which is in turn the first target platform of the T4P4S [4] P4 compiler. Since most bounded loop analysis problems are exponential in the bound, we choose an approach to predefine solution templates for some specific loops. For now, our purpose is to illustrate how to extend the probability model in Section III, so we use simplified models. We plan to evaluate and improve these models with real-world P4 compilers in future work.

A. Linear model for LPM lookup

First, we consider modeling the execution cost of the simple search algorithm: linear search. Since LPM is basically a search for maximum, we have to assume that the table is lexicographically sorted with decreasing mask lengths. It is unlikely for any real world application to use linear search for lookup: we feature it here because it is simple, fundamental, and we intend to demonstrate how to include loops in the presented model. Consider the pseudocode in Listing 2. First,

Listing 2 Pseudocode illustrating linear lookup

```

1  tbl := memory.open(tbl);
2  for(r in tbl){
3    cache.ensure(r);
4    b := r.match(hdr);
5    if(b.isMatch)
6      return b;
7  }
```

we read a cache line sized chunk of the table into memory, start the search loop and, in case we left the cached part of the table during the search, we cache the next chunk of the table. The average execution cost of this simple algorithm can be approximated using the following formula:

$$\begin{aligned}
& E[L(n, p, e, c_{M \rightarrow C}, c_m)] \\
&= \sum_{i=1}^n i G_p(i) (c_m + q(n, e) c_{M \rightarrow C}) \\
&= \frac{c_m + q(n, e) c_{M \rightarrow C}}{p}
\end{aligned} \tag{4}$$

In Equation 4, n stands for the number of entries in the match-action table, e for the size of the pattern to be matched, and – since, we do not have any information on arriving packets and tables – we can chose $G_p(i)$ to be the geometric distribution: $G_p(i) := (1 - p)^{i-1} p$. Here, $G_p(i)$ the probability of the exit condition (successful match of the packet) failing $i - 1$ times and succeeding the i th time, where p is the constant probability of the exit condition succeeding. Term c_m is the cost of matching an entry, $c_{M \rightarrow C}$ is the cost of caching a cache line sized chunk of memory, and $q(n, e)$ is the probability that caching must be performed in the i th iteration. Note, that we decided to make these members independent of the loop index.

We can approximate $q(n, e)$ using amortized analysis. The number of rows fit in the cache is $n_{\text{cache}} := \lfloor \frac{\text{size}_{\text{cache}}}{e} \rfloor$, and in the worst case, we need to perform caching $w := \lceil \frac{n}{n_{\text{cache}}} \rceil$ times. In average, we cache $\frac{w}{n}$ times per iteration.

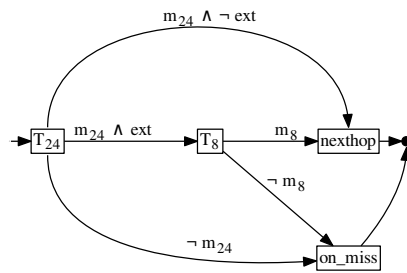


Fig. 1. A flowchart illustration of the DIR-24-8 algorithm

B. DIR-24-8 algorithm

DIR-24-8 was introduced by Gupta et al. [5] as a fast solution for LPM-based IP4 routing. In real networks most packets can be routed using just the first 24 bits: the basic idea is to take advantage of array indexing and cache efficiency by removing masks from the first 24 bits via prefix expansion and applying linear probing to match the last 8 bits.

The algorithm creates in RAM a table (named T_{24}) storing an entry for all (2^{24}) permutations of 24-bit addresses: each entry is either a pointer to T_8 (see below) or a 15-bit next hop address (another 1 bit denotes whether the value is a pointer or a next hop). Another table (named T_8) will match the remaining 8 bits of the address to the next hop. The set of all 24-bit addresses is a sequence between 0 and 2^{24} in base-256: that means T_{24} can be represented as an array with 2^{24} entries, each having a size of 16 bits. Thus, T_{24} will require $2 \cdot 2^{24}$ bytes (32 MiB) of space, while a section in T_8 corresponding to a 24-bit prefix will require at most $2^8(1 + 2) = 768$ bytes. The main advantage of the algorithm is that looking up the first 24 bits in T_{24} has constant cost (the address changed from base-256 to decimal, or some other base used for array indexing), and – if the entry contains a pointer – we only need to read an additional 768-byte parcel of T_8 , which easily fits even the smallest L1 caches.

ID	Probability	Cost
π_1	$P(m_{24} \wedge ext, m_8)$	$c(T_{24} \pi_1) + c(T_8 \pi_1) + c(nexthop)$
π_2	$P(m_{24} \wedge ext, \neg m_8)$	$c(T_{24} \pi_2) + c(T_8 \pi_2)$
π_3	$P(m_{24} \wedge \neg ext)$	$c(T_{24} \pi_3) + c(nexthop)$
π_4	$P(\neg m_{24})$	$c(T_{24} \pi_4)$

TABLE I
CHARACTERISTICS OF EACH PATH IN FIGURE 1

In this paper, we assume that T_8 lookups are performed using linear search: we expect practical implementations to use more efficient lookup schemes. Figure 1 depicts a schematic CFG of DIR-24-8. As in Section III, we calculate the expected value as a weighted sum of the CFG paths. Below, let $\pi_1 := m_{24} \wedge ext \wedge m_8$, $\pi_2 := m_{24} \wedge ext \wedge \neg m_8$, $\pi_3 := m_{24} \wedge \neg ext$, and $\pi_4 := \neg m_{24}$. For now, we assume m_{24} , ext , and m_8 are independent, and as such we will use the notation $r_8 := P(m_8|ext, m_{24})$, $r_{ext} := P(ext|m_{24})$, and $r_{24} := P(m_{24})$.

Let p_{24} and p_8 be the probability that an arbitrary entry is matching in T_{24} and T_8 respectively. Then $r_8 = 1 - (1 - p_8)^{2^8+1}$ and $r_{24} = 1 - (1 - p_{24})^{2^{24}+1}$.

It follows that $P(\pi_1) = r_8 r_{ext} r_{24}$, and $P(\pi_2) = (1 - r_8) r_{ext} r_{24}$, and $P(\pi_3) = P(m_{24}, -ext) = (1 - r_{ext}) r_{24}$, and $P(\pi_4) = (1 - r_{24})$. As a result, only the probabilities r_{ext} , p_8 , p_{24} need to be supplied as input.

$$\begin{aligned}
& E[\text{LPM}_{\text{DIR-24-8}}(p_{24}, p_8, r_{ext})] \\
&= P(\pi_1)(E(T_{24}|\pi_1) + E(T_8|\pi_1)) \quad + \\
&\quad P(\pi_2)(E(T_{24}|\pi_2) + E(T_8|\pi_2)) \quad + \\
&\quad P(\pi_3)E(T_{24}|\pi_3) \quad + \\
&\quad P(\pi_4)E(T_{24}|\pi_4) \quad + \\
&\quad (P(\pi_1) + P(\pi_3))\text{cost}(\text{nexthop}_9)
\end{aligned} \tag{5}$$

where

$$\begin{aligned}
E(T_{24}|\pi_1) &= \text{cost}(\text{binToDec}_4) + \text{cost}(\text{arrIdx}_2) \\
E(T_{24}|\pi_4) &= E(T_{24}|\pi_3) = E(T_{24}|\pi_2) = E(T_{24}|\pi_1) \\
E(T_8|\pi_1) &= E[L(2^8, p_8, 3, c_{M \rightarrow C}, \text{cost}(\text{match}_1))] \\
&= \frac{\text{cost}(\text{match}_1) + q(2^8, 3)c_{M \rightarrow C}}{p_8} \\
E(T_8|\pi_2) &= (2^8 + 1)(\text{cost}(\text{match}_1) + q(2^8, 3)c_{M \rightarrow C})
\end{aligned}$$

In the final formula, we only have to substitute runtime information for p_8 , p_{24} , and r_{ext} , and hardware or implementation specific information $\text{cost}(\text{binToDec}_4)$, $\text{cost}(\text{arrIdx}_2)$, $\text{cost}(\text{match}_1)$, $c_{M \rightarrow C}$, $\text{cost}(\text{nexthop}_9)$. Examples for these can be seen in Table II.

For T_{24} lookup, we assumed the cost consists of two substeps: transforming the address to an array index, and performing an array lookup in memory. As we modeled T_8 lookup using linear search, on π_1 , we use Equation 4 and prepare for an early exit. On π_4 , we use the information from π_4 that T_8 had to be read through in its entirety, i.e. the probability of the algorithm succeeding in the i th step is:

$$P(\text{exit}(i)|\pi_4) = \begin{cases} 1 & , \text{ if } i = 2^8 + 1 \\ 0 & , \text{ otherwise} \end{cases}$$

V. CASE STUDY

In this section, we demonstrate an application of the cost analysis procedure outlined earlier by examining how the cost of one LPM lookup changes for different parametrisations given data extracted from P4 source code. Here, we only have space to analyse the invocation `fib_lpm.apply()`, that will perform a longest prefix match on table `fib_lpm`. Using the data (partly utilized also in our earlier work [2]) in Table II to instantiate the formula in Equation 5, we obtained the graph in Figure 2. This plot exemplifies how increasing the cache size from 128B to 1024B results in decreasing execution costs. While L1 caches of this size are superseded today, translating DIR-24-8 to data larger than IP4 addresses also requires larger caches. We should also remember that real world implementations of DIR-24-8 are unlikely to use linear search for T_8 lookup). Larger cache means fixed sized data can be loaded in less memory reads, yet it is not evident how important is caching in the overall cost. By calibrating control event probabilities, we can observe that different program

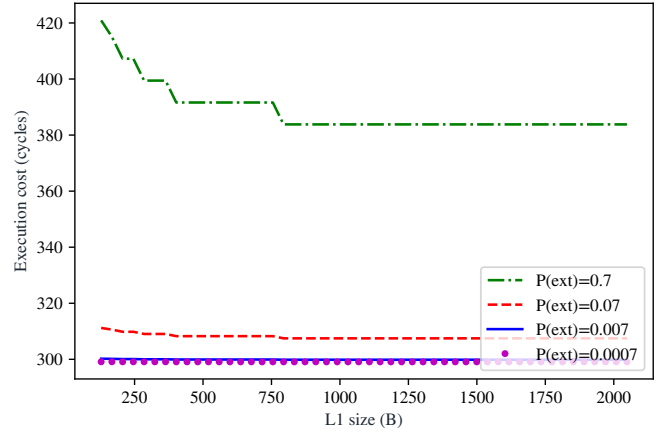


Fig. 2. Cost of `fib_lpm.apply()` over cache size and prefix lengths

paths gain more weight in the expected cost. In the figure, we experiment with increasing the probability of an address requiring T_8 lookup (*ext* event). According to Gupta et al. [5], 99.93% of prefixes in IP4 backbone routers in 1998 were at most 24 bits long, that is, the probability that a random address must be matched to a more than 24 bits long prefix were 0.0007. The plot confirms that for this kind of packet distribution DIR-24 is very efficient: the expected cost is just slightly above the cost of one memory read. As expected, the cost grows fast with every increase in T_8 lookup probability, suggesting that the worst case cost of the algorithm is very high. Note that if matching an entry costs only as much as 11 CPU cycles, reading through $2^8 = 256$ entries is already 2816 cycles, and we did not even factor in the required caching. For comparison, a 4.3GHz processor operating in a 10 Gigabit Ethernet network can spend only 288 cycles to forward a packet without risking buffer overflow [2]. Another property of DIR-24-8 illustrated by this plot is that cache size only becomes critical when T_8 lookups have a high probability (as T_{24} lookup only requires constant one memory read). Already, the cost difference between a 128B and a 1KB sized cache is non-negligible.

VI. RELATED WORK

In our earlier approach to P4 cost analysis [2], we syntactically transformed the code of P4 parsers into cost expressions. In the current paper, we utilized our earlier insights and continued this analysis with those of lookup tables, but our current algorithm is based on CFG traversal. Our main issue with the transformational approach was that computational costs of symbolic rewriting are hard to predict: even though it is important for exponential use cases such as ours.

P4-enabled switches can be considered soft real-time systems as there are timing requirements in addition to functional requirements: if the switch consistently fails to process packets before the network delivers a new packet, the packet buffer overflows and the switch fails out of the network. Davis et. al.

Symbol	Value	Meaning	Knowledge source
w	4B	CPU word length	Hardware configuration
$c_{M \rightarrow C}$	279	Cost of read from memory to cache	Hardware configuration
$c_{C \rightarrow R}$	5	Cost of read from cache to CPU register	Hardware configuration
$c_{R \rightarrow C}$	5	Cost of read from CPU register to cache	Hardware configuration
$cost(memCpy_v)$	$\lceil \frac{v}{w} \rceil (c_{C \rightarrow R} + c_{MOV} + c_{R \rightarrow C})$	Cost of copying a value	Implementation
$cost(binDec_v)$	$\lceil \frac{v}{w} \rceil (c_{C \rightarrow R} + c_{ADD} + c_{POW} + c_{R \rightarrow C})$	Cost of transforming IP4 address to array index	Implementation
$cost(arrIdx_v)$	$c_{M \rightarrow C} \lceil \frac{v}{size_{cache}} \rceil$	Cost of array lookup with v sized entries	Implementation
$cost(nextHop_v)$	$cost(memCpy_v) + (c_{C \rightarrow R} + c_{DEC} + c_{R \rightarrow C})$	Cost of <code>nextHop</code> action with v -bit destination port	Implementation
$cost(match_v)$	$\lceil \frac{v}{w} \rceil (c_{C \rightarrow R} + c_{CMP} + c_{R \rightarrow C})$	Cost of matching a v sized header to a table entry	Implementation
p_8	0.1	Probability of arbitrary T_8 entry matching	Inferred from table and the packet distribution. (Now arbitrary.)
p_{24}	0.1	Probability of arbitrary T_{24} entry matching	Inferred from table and the packet distribution. (Now arbitrary.)

TABLE II
EXAMPLE CONFIGURATION SPECIFICATION USED FOR LOOKUP COST ANALYSIS

[6] offers a classification and current survey of approaches for analysing timing requirements. In this classification, our work is considered static probabilistic timing analysis (SPTA).

A mature example of general SPTA theory can be found in Baier et. al [7]. Here, the authors use various Markov-chains for probabilistic analysis of semantic properties (such as state reachability), but also for calculating expected reward. While our work is more focused on application, in the future, we plan to discover the possibilities for improving fundamentals in this work.

A current example for dynamic timing analysis is Iyer et. al [8]. The authors developed Bolt, a tool based on dynamic instrumentation, that generates performance expressions from the C code of network functions. Performance expressions are terms (similar to the expected value formulas in our current work), containing performance critical variables (variables describing factors of hardware, implementation and execution environment that have high influence on performance). While we cannot directly compare this work to our work because of the difference in approach, the capabilities of the tool are astounding and serve as a goal for future improvement.

VII. CONCLUSION

Network developers utilizing P4 can significantly lower their operational costs by acquiring insights about the setup to be deployed as early as possible, preferably before deployment, before testing, and during development. In this work, we presented a cost analysis framework enabling refinement of P4 programs (in this context, considered as switch control flow specifications) as the continuation of our earlier work [2]. After presenting the technical background regarding P4 and match-action tables, we introduced a probabilistic model we believe is generic enough for analysing all relevant P4 setups, and yet easily extensible with new or better quality information. As an exercise in refining this model, we modelled P4 LPM lookup using the DIR-24-8 algorithm, and observed whether the expected value induced by the model conforms our expectations regarding this algorithm.

In future work, we would like to evaluate the feasibility of this model further regarding two aspects. The algorithm for

computing the expected value of a CFG we presented has exponential complexity in the number of branching nodes: yet, by utilizing compositionality and correspondence between CFG and expected value, we have some hope that the algorithm can be linearised in practice. The other aspect concerns applicability of this model to existing implementations of P4, such as T4P4S [4]. Even though the small case study here showed promising results, we have yet to see if the current model is ready for taking in real world implementation information, and if it can predict the observed cost of running P4 programs generated by actual P4 compilers.

REFERENCES

- [1] P. Bosshart, D. Daly, M. Izzard, N. McKeown, J. Rexford, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2656877.2656890>
- [2] D. Lukcs, G. Pongrcz, and M. Tejfel, "Keeping P4 Switches Fast and Fault-free through Automatic Verification," *Acta Cybernetica*, vol. 24, no. 1, pp. 61–81, May 2019.
- [3] Intel Corporation, "LPM Library, Chapter 24 in DPDK Documentation Programmers Guide," https://doc.dpdk.org/guides/prog_guide/lpm_lib.html, 2014, [Online; accessed 12-May-2019].
- [4] S. Laki, D. Horpácsi, P. Vörös, R. Kitlei, D. Leskó, and M. Tejfel, "High speed packet forwarding compiled from protocol independent data plane specifications," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, pp. 629–630. [Online]. Available: <http://doi.acm.org/10.1145/2934872.2959080>
- [5] P. Gupta, S. Lin, and N. McKeown, "Routing lookups in hardware at memory access speeds," in *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98)*, vol. 3, March 1998, pp. 1240–1247 vol.3.
- [6] R. Davis and L. Cucu-Grosjean, "A Survey of Probabilistic Timing Analysis Techniques for Real-Time Systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: <https://hal.inria.fr/hal-02158973>
- [7] C. Baier and J.-P. Katoen, *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.
- [8] R. Iyer, L. Pedrosa, A. Zaostrovnykh, S. Pirelli, K. Argyraki, and G. Candea, "Performance contracts for software network functions," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 517–530. [Online]. Available: <https://www.usenix.org/conference/nsdi19/presentation/iyer>

Accuracy of Unit Under Test Identification Using Latent Semantic Analysis and Latent Dirichlet Allocation

1st Matej Madeja

*Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
matej.madeja@tuke.sk*

2nd Jaroslav Porubän

*Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
jaroslav.poruban@tuke.sk*

Abstract—Identification of unit under test (UUT) from a test is often difficult and requires wider source code comprehension. By automating this process it would be possible to support the program comprehension and reduce software maintenance process. In this paper the Latent Semantic Analysis (LSA) and the Latent Dirichlet Allocation (LDA) were used which proved to be inaccurate in the UUT identification. The experiment was conducted on 5 popular projects where 1,093,730 similarity results were obtained. It was found out that the best topic number for the LSA model is from 7 to 10, the LDA model had big differences in this value, so it was not possible to define a stable value. The best UUT identification accuracy compared to manual testing has been obtained with the LSA model with result of 7.63% success, where documents were preprocessed using words splitting based on naming conventions and Java keywords removal. The accuracy of the LDA model was almost zero. Further 8 manual identification errors were discovered during the experiment.

Index Terms—program comprehension, latent semantic analysis, latent dirichlet allocation, github mining, unit under test

I. INTRODUCTION

Natural language processing (NLP) attempts to reduce the barriers in computer-to-human communication [1]. This process involves the correct text analysis, then the determination of its semantics and the execution of the required action. During the implementation of the program source code the programmer writes statements in the prescribed syntax of a non-natural language but many times expresses thoughts that are important to facilitate program comprehension in the future – either for himself/herself or for other programmers. These thoughts are expressed mainly by following naming conventions (e.g. Java Coding Style Guide [2]) which task is to simplify the representation of the code in the problem domain, e.g. by semantically correct naming of classes, methods, variables, etc. Based on the research by Butler et al. [3] it can be seen that the names of the identifiers have a significant impact on the information mining from the source code and its comprehension by the programmer.

In our previous research [4] we focused on word frequency analysis between test and production classes in 5 popular Android projects on Github¹ and general testing practices. Because the tests can be considered always up-to-date documentation of the production code we found out that the words used by the programmers in the tests and the production code are very similar. It was also found that 49% of the analyzed test titles included the full name of a particular unit under test (UUT) and even 76% the partial name. The production and test method bodies also used similar vocabulary. A lot of vocabulary is placed in comments in the form of natural language.

From the above it is possible to assume that each source code file uses its own vocabulary. The used words are not in the form of sentences so it is not possible to search for semantics between words within the final set of words (sentence). But each document contains a set of words that together characterize a whole, e.g. in Java the most common a single file will represent a class.

A general problem during the test code comprehension is the UUTs identification, especially when one test class tests multiple production classes. According to McGlaufflin [5] in Java one production class should be tested by only one test class and the programmer is led to this convention also using an integrated development environment (IDE) tool. If this convention is followed it can be assumed that the test and production classes will have similar vocabulary.

The aim of this paper is to use 2 models of natural language processing (NLP) techniques: *Latent Semantic Analysis* (LSA) and *Latent Dirichlet Allocation* (LDA). These models are some of the least computationally complicated in the NLP field and could greatly help identify UUTs based on the vocabulary of test, thereby simplify program comprehension and prevent faults. In this paper the following research questions are discussed:

¹<https://github.com/>

- RQ1:** Is there a general topics number for processing source code files without the need of searching it?
- RQ2:** How exactly can UUT be identified from the test class vocabulary?
- RQ3:** How to preprocess source code documents for model training to the obtain best results?

In the Section II we briefly describe usage basics of LSA and LDA models to process natural text without mathematical details, since they are not important to the problem this paper deals with. The Section III describes the programming language selection, used libraries and data preparation for processing. The results are described in the Section IV and at the end of the paper threats to validity, related work, conclusions and future work are discussed.

II. LSA AND LDA MODELS

Both models are information retrieval (IR) algorithms that expect vectors as input, mainly because their nature is mathematical operations involving matrices. The input strings are therefore represented as vectors and this type of representation is called *Vector Space Model*. Based on these vectors a particular NLP model can make predictions. The aim of these algorithms is to train the model from the input data to minimize the occurrence of prediction errors.

There are different representations of text as a vector. The most straightforward representation is *bag-of-words* (BoW) and it is an orderless document representation, so only the counts of the words matter. This leads to loss of word order, syntactic relations, or morphology [6]. However, in our case most of the input data except comments will not have the form of natural text, so this is not critical. For most IR algorithms the frequency of word occurrence is sufficient for calculation. A slightly more sophisticated model is *tf-idf* (*term frequency-inverse document frequency*), which tries to encode two different kinds of information - term frequency and inverse document frequency [7]. Term frequency (*tf*) is the number of times the word appears in a document. It is possible to extend these models, for example using topic models. Both LSA and LDA can use *bag-of-words* model to obtain term-document matrix.

In the following subsections selected models without mathematical details are described. A detailed explanation of both models can be found in [8] and [9]. Cvitanic et al. [10] discuss models' differences in more detail.

A. Latent Semantic Analysis

LSA is an indexing and IR method that uses *Singular Value Decomposition* (SVD) to identify relationships between words in an unstructured text. The model is based on the assumption that words used in the same context have a similar meaning [8]. By extracting terms from the document's body it seeks to create relationships between individual documents. It is important to choose a right number of topics to generate because if too many topics are requested for a short document the algorithm returns also words that should not determine the resulting topic of the document and vice versa.

B. Latent Dirichlet Allocation

The model considers each document as a set of topics which characterize it [9]. Each topic consists of a set of words in a certain proportion. Based on the number of topics required the model attempts to rearrange the topics distribution within the documents to achieve the best composition. It is also very important to determine the right number of topics that the algorithm returns.

C. Number of topics

To evaluate the quality of a trained model, which is significant for results of both considered models, can be determined by topic coherence that is a measure used to evaluate topic models. The topic coherence is applied to the top N words from the topic and it is defined as the average/median of the pairwise word-similarity scores of the words in the topic [11]. A good model will generate coherent topics with high coherence scores. Good topics are those that can be described by a short label.

III. METHOD

The experiment was conducted on 5 popular Android projects from our previous research [4]. In order to know the success rate of particular NLP models in identifying a UUT from a test it is necessary to establish a link between the test and the production classes. Since we performed a manual analysis of 617 tests in [4] we can partially use the collected data for this experiment. We assume that manually created links are correct. The source codes of considered projects (see Table I) are from February 2019 to preserve consistency with manually collected data. The projects were selected on the assumption that the most popular projects will include tests (see more in [4]).

TABLE I
GENERAL STATS OF MANUALLY ANALYZED DATA. [4]

Project	Prod. classes	Prod. methods	Test classes	Test methods
plaid	37	71	39	180
ExoPlayer	49	98	53	323
Android-Clean Architecture	17	22	17	29
shadowsocks-android	6	7	6	8
iosched	16	40	16	77
SUM	125	238	131	617

The Table I shows that in most cases the convention that one UUT (production class) is tested by one test class has been fulfilled, so in 125 cases we can clearly establish the expected connection between the test and production class. For tests that test multiple production classes the most tested production class will be considered as correct UUT.

A. Programming language and library

For projects' source code analysis we chose Python language which is great for processing computationally difficult tasks. Also availability of *gensim* [12] library for this language

was crucial. The library is very popular in the NLP field and according to its author Řehůřek [13] *gensim* is the most robust, efficient and hassle-free piece of software to realize unsupervised semantic modeling from plain text.

B. Documents preparation

All analyzed projects are built on the Android platform and implemented in Java and/or Kotlin. For each project we recursively searched for files having the `.java` or `.kt` extension. Kotlin is designed to interoperate fully with Java so they use similar programming conventions and it is also suitable for our analysis. Of course, only the files of the project were included in the analysis, i.e. without dependencies and platform software development kit files. We already knew the names of the test classes from [4] so we divided the particular files into test and production set. The content of production classes served for model training and content of test classes for searching similarity.

File preprocessing was the same for both test and production classes. From the content of each file new line characters have been removed and result was saved in a project-depended training file. One line in this file represented one document for further processing.

Since the result is highly dependent on the quality of the input data and is governed by the idiom "garbage in, garbage out", it is very difficult to assume in a non-natural text how it should be properly preprocessed. To find out how to prepare input documents in the best way (RQ3), we incrementally created 5 versions of document preprocessing:

- 1) *Full version* - original file version, removed only `\n` chars.
- 2) *Word split* - all camelCase or snake_case words has been split. Words out of base conventions, such as `ORMLite`, remained unsplit.
- 3) *Removed Java keywords* - all Java keywords have been discarded.
- 4) *Removed comments* - multi- and one-line comments discarded.
- 5) *Removed imports* - all Java imports removed.

Incremental preprocessing means that for example in the 4th iteration also *Word split* and *Removed Java keyword* have been included. Another preprocessing of documents that applied to all iterations was the removal of frequently occurring English words using *nlk* library, such as *and*, *a*, *the*, etc. At the same time stemming over the documents has been executed, where inflected or sometimes derived words to their word stem have been reduced, e.g. *cars* to *car*. The last step was to remove words that occurred only once in the corpus of training documents to eliminate their negative impact on results.

C. Model training

Creating a *bag-of-words* representation in the form of a dictionary (*id + word* pair) and creating a corpus of sparse vectors was relatively easy using the functions of *gensim*. As mentioned in the Section II, choosing the right number of topics is a much bigger challenge. Since the number of topics

is dependent on the nature of the documents we searched for the highest coherence value for each iteration and project.

Training a large number of models is very time consuming task so in the early stages of the experiment we tried to obtain approximate range to try in. After multiple tests we decided to calculate coherence values from 7 to 50 for each model, project and iteration, and the model with the highest value has been chosen for the analysis.

To train the LSA model only the necessary parameters were supplied - number of topics, dictionary (BoW) and corpus (of vectors). For the LDA model training we also set the *alpha = auto* parameter which means the model learns asymmetric prior from provided corpus. From *alpha* attribute LDA model computes *theta* that decides how the topic distribution is drawn. The last special parameter set for the LDA model was *passes = 20* which express the number of passes through the corpus during training. In terms of statistics, more training means statistically more accurate results.

D. Evaluation of document similarity

When using LSA or LDA we focused on a single aspect of possible similarities, i.e. on apparent semantic relatedness of their texts (words), just a semantic extension over the boolean keyword match. Modern search engines also take into account random-walk static ranks, hyperlinks, etc. *Gensim* basically uses *cosine similarity* [14] to determine the similarity of two vectors and it is a standard measure in *Vector Space Modeling*.

The search query was made up of the content of a test class. We created *bag-of-words* for each document (test body) and converted it to the corresponding LSI/LSA space. Subsequently, an index has been created from the trained model against which the query was evaluated. Similarities to all production classes were calculated and we obtained the result as (*document_no*, *similarity_value*) pairs where *similarity_value* $\in (-1, 1)$. The greater *similarity_value* the more similar document. Every *document_no* has been paired with value stored in a relation database created during training document preparation to identify particular production class.

IV. RESULTS

Altogether we analyzed 2221 production and 168 test classes in five projects (see Table II). In five iterations of document preprocessing a total of 1,093,730 similarity results between the tests and the production source code have been obtained. In the Table II also statistics of mean search time in prepared index are presented, as can be seen, searching in LDA index is mostly a little bit faster.

A. Optimal topic number and train times

Model training lasted the longest time especially because of searching for the optimal topic number. The tests were performed on 12-core *Intel Core i7-5820K* CPU with 12GB RAM and *Debian GNU/Linux 9* installed. Nevertheless, analysis was performed in one thread only to simulate the use in real environment, e.g. in the background of IDE. The average search times for the highest coherence value can be seen in the

TABLE II
GENERAL DOCUMENT STATISTICS IN ANALYZED PROJECTS.

#	Project	Number of files		Mean search time in index (1 query)	
		prod.	tests	LSA	LDA
1	plaid	679	39	49.73 ms	43.97 ms
2	ExoPlayer	954	53	78.52 ms	88.33 ms
3	Android-Clean Architecture	99	17	7.65 ms	3.53 ms
4	shadowsocks-android	157	6	11.67 ms	10.00 ms
5	iosched	332	16	18.89 ms	13.33 ms

Table III. The differences in individual LDA iterations were minimal, i.e. in tens of seconds, with the LDA model the differences were even a few minutes. The greatest decrease of training time in a particular NLP model was recorded in the 4th iteration when comments were removed. By removing them a lot of the training data have been lost so finding the best coherence value in the 4th iteration was faster: 2.17 times for LSA and 2.83 times for LDA. Despite the increased speed a large amount of potentially natural text in the source code has been lost.

TABLE III
AVERAGE TIMES OF SEARCHING BEST COHERENCE VALUES PER PROJECT IN MINUTES.

Project #	1	2	3	4	5
LSA	3.96m	4.46m	0.97m	2.03m	1.36m
LDA	21.31m	32.66m	2.26m	6.84m	7.18m

- to identify particular project pair with Table II.

To answer **RQ1** the Table IV was created. As can be seen the mean value for LSA is relatively stable. Although the maximum deviation of the LSA value is 40, this situation occurred in only one case, i.e. it was just an exception that could be neglect. For the source code analyses using the LSA model it is therefore possible to use a relatively stable value of topic numbers in the range 7–10.

TABLE IV
TOPIC NUMBER MODE VALUE, TOPICS NUMBER DIFFERENCES AND MODEL TRAINING TIME WITH BEST COHERENCE VALUE PER PROJECT.

Subject	Metric	Model	Project #				
			1	2	3	4	5
Topic No.	mode	LSA	7	7	7	10	10
		LDA	-	-	14	12	19
Topic number difference	min	LSA	0	0	0	0	0
		LDA	1	1	0	0	0
	max	LSA	1	6	40	6	5
		LDA	11	20	25	17	23
Best model train time (s)	min	LSA	0.09	0.14	0.02	0.03	0.03
		LDA	8.92	11.27	1	2.88	3.82
	avg	LSA	0.14	0.19	0.02	0.04	0.05
		LDA	12.35	24.31	1.29	4.27	5.3
	max	LSA	0.2	0.3	0.02	0.05	0.08
		LDA	17.86	44.74	1.5	8.37	7.23

- to identify particular project pair with Table II.

When using LDA the selected topic numbers were very diverse as the model is less stable. This can be obtained in

multiple model training with the same data when the results vary slightly. That's why topic numbers are more diverse than in LSA. Most often the topic number for a project was in similar values and in the difference range of 10 units. However, the differences between the particular projects were large and based on this data it is not possible to determine the recommended topic number for LDA.

B. Accuracy of UUT identification

Since we assume that manually identified UUTs are correct it is possible to determine the accuracy of a particular model based on the order of production class in the search result. The Figure 1 shows the frequencies in the search queries for manually labeled production classes as UUT.

As can be seen, LSA performed much better than LDA. The LSA is based on the frequency of words in the documents and as was found in [4], the words between the test and UUT are very similar which positively influenced the result. The accuracy of the LDA model was very low, in the first five results the correct UUT appeared only 2 times. Although the LSA achieved 82 correct UUTs in the first five results for all iterations it is still only 13.33% success rate which is considerably inadequate. In response to **RQ2** from our results, only 5.20% of UUTs were marked correctly (all iterations) and solely by the LSA method. In the results it is necessary to take into account the fact that for the 6 test classes, which tested multiple production classes at once, the most tested class was chosen as the correct UUT (discussed in Section III).

A more detailed look at the best search results in each iteration is needed to respond the **RQ3** (see Table V). As can be seen, *word split* and *removal of java keywords* (I2 + I3) has the greatest impact on the accuracy of the results. Our expectation was that when comments are removed the results will get worse because there is a potential for sole natural language in the comments. In the Table V it can be seen that removing comments (I4) and imports (I5) had a negligible impact on accuracy. It also shows that the meaning in the code is most often expressed directly in the names of the identifiers, i.e. class, methods and variables names. Using word splitting were obtained the most accurate results, it was the fastest iteration in terms of model training, finding the best coherence value and search in the index (Table II).

TABLE V
POSITION FREQUENCY FOR FIRST 5 POSITIONS OF ANALYZED MODELS.

Iteration	Position frequency in the search											
	LSA						LDA					
	1	2	3	4	5	Σ	1	2	3	4	5	Σ
I1		1				1			1			1
I2	4	1	3	5		13						
I3	10	8	2	4	1	25			1			1
I4	9	6	4	1	2	22						
I5	9	6	2	3	1	21						
Σ	32	22	11	13	4		0	0	2	0	0	

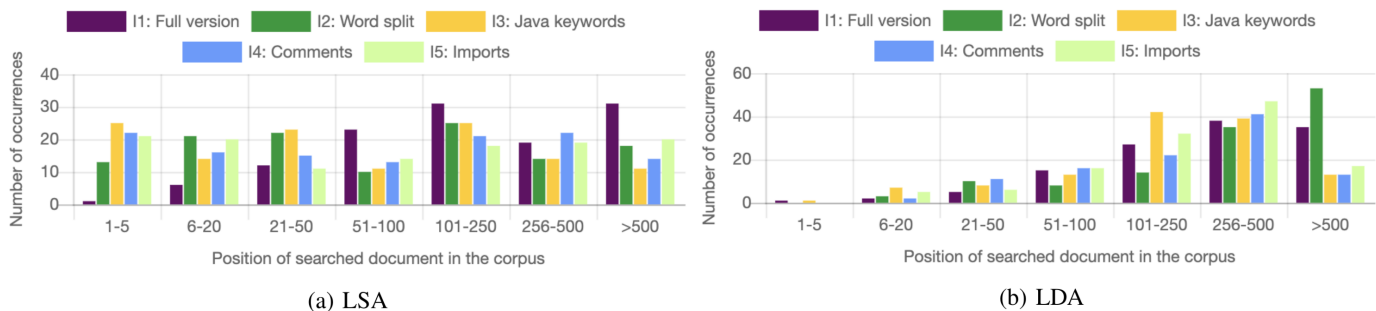


Fig. 1. Position frequency of manually identified UUT for LSA and LDA models.

C. Detected errors of manual identification

In the experiment, we unexpectedly found that 8 classes manually labeled as UUT were not in the corpus. Manual testing was done in *Android Studio* IDE where we used references created directly by IDE. After a more detailed analysis we found that production classes that were not in the corpus were incorrectly labeled. Incorrect identification was due to references to generated source code that were not present in the file system of the project without run. The method can be therefore used to prevent such errors.

V. THREATS TO VALIDITY

Comparison the accuracy of LSA and LDA was relied on the fact that the manual identification of UUTs was performed correctly. If an error in manual identification happened this could have a negative effect on the results reported in this paper. The analysis was performed on only 5 popular Android projects and no other projects were included, e.g. less popular, proprietary, etc. At the same time, projects in other languages have not been analyzed and a particular language can affect the corpus of words by using naming conventions or language syntax.

While the LDA method is more accurate than the LSA (claim from the official description of the method, not from our results), on the other hand, this method also shows slight differences in document comparisons when training the LDA model multiple times with the same data, indicating some inaccuracy but statistically friendly.

The preprocessing of documents (the *garbage in, garbage out* idiom) has also a huge impact on the results. How to prepare source codes for such analysis was also one of the research questions of this paper. There exist a thread to validity because not all possible document preparation could be tried out.

As mentioned in Section II the choice of topics number also has a big impact on the accuracy of the methods. Despite finding the best value for this parameter a search range of 7 to 50 may not be sufficient. Also the use of coherence value may not be reliable at all times and there is no general recommendation on how to accurately determine this parameter, so there is no guarantee that the best values have been chosen with respect to the input data.

VI. RELATED WORK

The most similar research on improving of program comprehension was done by Maletic et al. [15], [16]. In their conclusions they argue that the LSA model can assist in supporting some of the activities of the program's comprehension process. However, they only analyzed one project in C, in our case a larger sample and Java and/or Kotlin languages are considered. They analyzed 269 files, we analyzed on a sample of 2221 files in all projects without 168 test files used as queries on index. They created code clusters of similar files trying to make it easier for the programmer to find related parts of the program. In our case we focus on the relationship between the test and the production class which can even be written in another language (e.g. tests in Java, production code in Kotlin; see *plaid* project) and assume that the UUT test will have more common vocabulary as 2 different classes.

Another type of research was performed by Thomas et al. [17], [18] in 2014 who mined software repositories using topic models to simplify the understanding of software changes during software evolution, especially for stakeholders. Although their experiments have not been verified on real stakeholders the results show that extraction of topics is sufficient and should therefore have a positive impact on simplification of understanding. In our research we focus more on developers, analyzing the source code and relationships within it.

Asuncion et al. [19] proposed an automated technique that combines traceability with topic modeling. They record traceability links during the software development process and learns a probabilistic topic model over different artifacts. From collected data they are able to categorize artifacts and create topical visualisations of particular system. They implemented several tools that support data collecting during software evolution. In our case, we are still in the early stages, so we found out whether it is beneficial to look for similarities between the test and the production code using NLP techniques.

VII. CONCLUSIONS AND FUTURE WORK

In this paper the use of 2 NLP methods for UUT detection in popular open-source Github projects is discussed. Previous research shows that using these methods for source code

analysis can positively affect the program comprehension of developers.

Five popular Android projects were used for the analysis which included 131 test and 2221 production classes in total. In experiment *Latent Semantic Analysis* and *Latent Dirichlet Allocation* models have been used which are able to evaluate the similarities between documents. The source code of the production classes was used to train the models and the content of the test classes were used for search queries. In the paper is described how to find the best value of topic number. The main objective was to find out how exactly these models can identify unit under test against manual identification. The experiment was conducted in five iterations and for every iteration training and search documents have been modified in different ways to determine how to preprocess source code documents for these models to get more accurate results.

It was found that a range of topic number from 7 to 10 is suitable for the LSA model and the choice of topic number was very varied for the LDA model, so a generally sufficient value could not be found. The accuracy of the LDA model was only 5.20%, after taking into account the small deviation only 13.33% records were found in the top five search results. The accuracy of the LDA model is almost zero, so it is not at all suitable for UUT identification. The fact that the LSA model was more successful in the experiment corresponds to the results of our previous research [4] that the vocabulary used in the test and UUT was often similar. The LSA method could therefore be a partial complement for more accurate UUT identification but not absolutely reliable.

The best results in terms of searching for the best coherence value, model training time, searching for similarity and UUT identification were recorded for 3rd iteration with the accuracy of 7.63%, taking into account the deviation of up to 5 records 19.08%. The mentioned iteration included: removing new lines; splitting words using camelCase and snake_case convention; removing Java keywords; removing English general words; stemming. At the same time, the class lookup notified us to 8 mislabeled UUTs during manual testing in which dynamically generated classes were identified as UUTs.

Despite the considerable failure of the methods the experiment also needs to be performed outside of Android projects and on a different sample of projects, e.g. in terms of length of maintenance, number of tests, language, etc. For example *Gherkin* which contains more of natural text the results could provide better results. In the future we will also look at comparing the results with other UUT identification techniques, e.g. observation of code co-evolution, helper methods, or naming conventions.

ACKNOWLEDGMENT

This work was supported by project VEGA No. 1/0762/19: Interactive pattern-driven language development.

REFERENCES

- [1] C. D. Manning, C. D. Manning, and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.

- [2] A. Reddy *et al.*, “Java™ coding style guide,” *Sun Microsystems*, 2000.
- [3] S. Butler, M. Wermelinger, Y. Yu, and H. Sharp, “Mining java class naming conventions,” in *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, Sep. 2011, pp. 93–102.
- [4] M. Madeja and J. Porubán, “Tracing naming semantics in unit tests of popular github android projects,” vol. 74, 2019. [Online]. Available: <https://www2.scopus.com/inward/record.uri?eid=2-s2.0-85071071510&doi=10.4230%2fOASICS.SLATE.2019.3&partnerID=40&md5=6f5044423719630d3eaa7ed15b351def>
- [5] B. McGlauffin, *Java Unit Testing Best Practices: How to Get the Most Out of Your Test Automation*. DZone Technical Library, 05 2019. [Online]. Available: <https://dzone.com/articles/java-unit-testing-best-practices-how-to-get-the-mo>
- [6] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010, vol. 520.
- [7] D. Hiemstra, “A probabilistic justification for using tfidf term weighting in information retrieval,” *International Journal on Digital Libraries*, vol. 3, no. 2, pp. 131–139, Aug 2000. [Online]. Available: <https://doi.org/10.1007/s007999900025>
- [8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.
- [9] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [10] T. Cvitanic, B. Lee, H. I. Song, K. Fu, and D. Rosen, “Lda v. lsa: A comparison of two computational text analysis tools for the functional categorization of patents,” in *International Conference on Case-Based Reasoning*, 2016.
- [11] J. H. Lau, D. Newman, and T. Baldwin, “Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality,” in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 530–539.
- [12] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, <http://is.muni.cz/publication/884893/en>.
- [13] R. Řehůřek, *About Gensim*, 07 2019. [Online]. Available: <https://radimrehurek.com/gensim/about.html>
- [14] A. Huang, “Similarity measures for text document clustering,” in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand, vol. 4, 2008, pp. 9–56.
- [15] J. I. Maletic and A. Marcus, “Using latent semantic analysis to identify similarities in source code to support program understanding,” in *Proceedings 12th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2000*, Nov 2000, pp. 46–53.
- [16] J. I. Maletic and N. Valluri, “Automatic software clustering via latent semantic analysis,” in *14th IEEE International Conference on Automated Software Engineering*. IEEE, 1999, pp. 251–254.
- [17] S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein, “Studying software evolution using topic models,” *Science of Computer Programming*, vol. 80, pp. 457–479, 2014.
- [18] S. W. Thomas, “Mining software repositories using topic models,” in *Proceedings of the 33rd International Conference on Software Engineering*. ACM, 2011, pp. 1138–1139.
- [19] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, “Software traceability with topic modeling,” in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, vol. 1. IEEE, 2010, pp. 95–104.

Comparison of Predictive Statistical Learning Accuracy with Computational Intelligence Methods

Dusan Marcek
 Institute of Informatics
 Silesian University in Opava
 Opava, Czech Republic
 dusan.marcek@fpf.slu.cz

Abstract— Forecasting of high-frequency economic time series data is a complex problem, which has benefited from recent advancements and research in machine learning. To forecast time series data, two methodological frameworks of statistical and computational intelligence modelling are considered. The statistical methodological approach is based on the theory of invertible ARIMA (Auto-Regressive Integrated Moving Average) models. The Computational Intelligence (CI) models are based on neural networks (NN) and Support Vector Machines (SVM). The main work of this study is to compare the predictive accuracy level of the statistical methodological approach with NN and SVM on the large data set. We evaluate statistical ML (Maximum Likelihood) learning method, Back-Propagation (BP) and genetic algorithms (GA) for half-hourly 1-step-ahead electricity demand prediction using Australian electricity data. We showed that all ARIMA, NN, SVM models as prediction methods are reasonable and acceptable for use in forecasting systems that routinely predict values of variables in competitive energy markets.

Keywords— ARIMA models, SVR, Neural networks, Learning algorithms, Roulette wheel.

I. INTRODUCTION

Electricity demand prediction is very important for the reliable and efficient operation of power systems. We consider predicting the electricity demand half hour ahead from previous half-hourly demands. This type of prediction is used for two main purposes: (1) to make decisions about dispatching generators and setting the minimum reserve during the daily operation of power systems and (2) to provide information to electricity market participants for their bidding in competitive energy markets. In both cases the goal is to ensure reliable electricity supply while minimizing the

There are three main groups of approaches for electricity demand prediction: the traditional causal (econometric), time series [1] and the more recent computational intelligence method, such as NN, SVM. In this paper we develop and compare the performance of the classic (that is of the perceptron type) RBF (Radial Basic Function) NN and RBF NN trained by GA with ARIMA models. The goal of this paper is to illustrate that the two distinct approaches, i.e. statistical models and computational networks may be used for financial high-frequency time series modelling.

The paper is organized as follows. In Section II we briefly describe the variants statistical ARIMA models (ARMA and seasonal ARMA model) and SVR model. In Section III we present the data, conduct some preliminary analysis of the time series and demonstrate the forecasting abilities of

classic/seasonal ARMA and SVR models. Section IV describes design and application of RBF NN trained with BP and GA algorithms. Section V presents results and empirical comparison. Section VI briefly concludes.

II. STATISTICAL TIME SERIES MODELS

A. ARIMA time series models

ARIMA time series model belongs to the group of Box and Jenkins [2] methods. The Auto-Regressive (AR) process is a linear combination of previous values, the Moving-Average (MA) process is a linear combination of previous errors, 'I' is an operator for differencing a time series. An ARMA(p, q) model of orders P and Q is defined by

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (1)$$

where $\{\phi_j\}$ and $\{\theta_j\}$ are the parameters of the autoregressive and moving average parts respectively, and ε_t is white noise with mean zero and variance σ^2 . We assume that ε_t is normally distributed, i.e. $\varepsilon_t \sim N(0, \sigma^2)$.

B. Seasonal ARMA Models

An extension of the ARMA process is a pure seasonal model abbreviated as ARMA(P, Q) $_s$ process in the form

$$y_t - \lambda_1 y_{t-s} - \lambda_2 y_{t-2s} - \dots - \lambda_p y_{t-ps} - \varepsilon_t = \gamma_1 \varepsilon_{t-s} + \gamma_2 \varepsilon_{t-2s} + \dots + \gamma_Q \varepsilon_{t-Qs} \quad (2)$$

where $\{\lambda_i\}$ are the seasonal autoregressive parameters, $\{\gamma_j\}$ are the seasonal moving average parameters and the subscripts s denote nonzero parameters that are integer multiple of s . The pure seasonal models defined by (2) are often not realistic since they are completely decoupled from each other. That is, (2) represents s identical but separate models and we need to take into account the interactions or correlations between the time series values within each period. This can be done by combining the seasonal and regular effects into a single model. We will use a multiplicative seasonal autoregressive integrated moving average process of period s (SARIMA(p, d, q)(P, D, Q) $_s$), with regular and seasonal AR orders p and P , regular and seasonal MA orders q and Q , and regular and seasonal differences d and D . In typical application, $D = 1$, the model is defined in Section III.

C. SVR model

Support Vector Regression (SVR) model is an extension of the SVM algorithm for numeric prediction and is

successfully applied in functional approximation. When used with kernel functions it can create complex nonlinear decision boundaries while reducing the computational complexity.

The initial hypothesis for choosing the shape of the SV regression model is the hypothesis that wages show inertia, which can be expressed in a simple causal model in the form of $y_t = \varphi y_{t-1} + \varepsilon_t$, where ε_t is the white noise random component. By taking this hypothesis into account in the SVR Model, its initial shape may take the following form

$$\begin{cases} f(\mathbf{y}, \mathbf{w}, b) = K(\mathbf{y}_i, \mathbf{y}_j) \mathbf{w} + b & \text{or} \\ f(\mathbf{y}, \alpha, b) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{y}_i, \mathbf{y}_j) + b. \end{cases} \quad (3)$$

where $K(\mathbf{y}_i, \mathbf{y}_j)$ are relevant kernel function, $\mathbf{y}_i, \mathbf{y}_j$, are training data, b is a real constant (bias). The real constants are obtained from the solution of the following quadratic programming (QP) problem [3].

$$\begin{cases} \max L(\alpha, \alpha_i^*) = \\ -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \psi(\mathbf{x}_i^T \mathbf{x}_j) - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \end{cases} \quad (4)$$

subject to constrains

$$\sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0 \quad (5)$$

$$0 \leq \alpha_i^* \leq C, \quad i = 1, n \quad (6)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, n, \quad (7)$$

where L is the Lagrangian with Lagrange multipliers α_i, α_i^* .

In the SV regression, to estimate its parameters the user must further choose some attributes that affect their estimates. These are the following attributes: measure of error approximation (Loss Function ε), the regularization and weights vector norm C , kernel function K and its degree (for further details see [4, 5]).

To evaluate the predictive accuracy of forecasting models, we used following statistical summary measures of a model's forecast accuracy defined as:

The Root Mean Square Error (*RMSE*) is

$$RMSE = \sqrt{MSE}. \quad (8)$$

The Mean Absolute Error (*MAE*) is

$$MAE = \sum_{i=1}^n |e_i| / n. \quad (9)$$

The Mean of the Absolute Percentage Error (*MAPE*)

$$MAPE = \frac{1}{n} \sum_{i=1}^n |e_i / y_i| 100 \quad [\%], \quad (10)$$

where e_t is the forecast error in time period t , and n is the number of observations in the estimation period.

III. AN APPLICATION OF SEASONAL ARIMA AND SVR MODELS

To illustrate the statistical methodology, consider half-hourly electricity demand data for the state of New South Wales in Australia for June, July and August 2010. Electricity demand data recorded at half-hourly intervals shows two main cycles: daily and weekly, see Fig. 1(a). The daily pattern shows that the demand is lowest at 4:30am and then reaches its first maximum at 9:30am and its second maximum at 6:30pm, in agreement with the human routine. The weekly pattern shows that he same days of the week (e.g. Mondays) have similar demand profiles. The weekly and daily cycles are 336 and 48 half-hour periods, respectively.

To build a forecast model the time series data was split into training and validation data set. We use the data for June and July as training data (i.e. for feature extraction, estimation of the parameters and building the prediction models) and the data for August as *testing data*. (i.e., for evaluating the performance of the prediction models).

The primary tool used in identification process is Auto Correlation Function (ACF). Actually, the theoretical ACF is unknown and must be estimated by the sample ACF. Fig. 1(b) shows the sample autocorrelation function for the training data; values close to 1 and -1 indicate high positive and negative linear correlations and values close to 0 indicate lack of correlation.

We extracted autocorrelation features that were shown to capture the daily and weekly cycles. We applied multiplicative seasonal ARIMA model which can be expressed as $ARIMA(p, d, q)(P, D, Q)_s$. After model identification, selection by using the Akaike criterion [6], the best fit model was specified as multiplicative seasonal $SARIMA(6, 1, 1)(1, 1, 1)_{48}$ process expressed in the following form

$$y_t (1 - \lambda_1 B - \lambda_2 B^2 - \dots - \lambda_6 B^6) = (1 - \gamma B^{48}) \varepsilon_t \quad (11)$$

where B is the backward-shift operator defined as $B = 1 - \nabla$ and $\nabla y_t = y_t - y_{t-1}$, $\nabla^d = (1 - B)^d$, $\nabla_s^D = (1 - B^s)^D$, where ∇ are so called difference operators. For details see [7].

The parameters $\{\lambda\}$ and $\{\gamma\}$ of the model (11) were estimated by ML procedure. Calculation of the MAPE value for validation data set was performed by e-views (<http://www.eviews.com>) software.

The SVR application was conducted using the variables and data sets as the statistical model above. In this study, according to the previous result, polynomial kernel was used with the measure of error approximation, $\varepsilon = 0.001$, and the regularization parameter $C = 1$, which influences a trade-of between an approximation error and weights vector norm. The prediction of half-hourly electricity demand data for the state of New South Wales in Australia has been done using *WEKA* software [8].

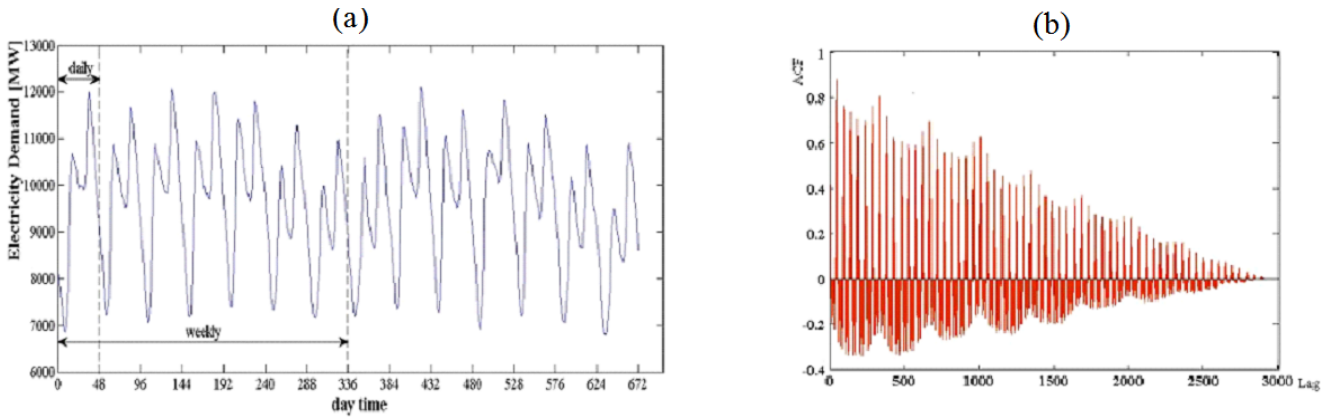


Fig. 1. (a) Electricity demand data for 2 consecutive weeks, (b) Autocorrelation function for the training set

IV. NEURAL NETWORK APPROACH

Neural networks can be understood as a system which produces output based on inputs the user has defined. It is important to say that user has no knowledge about internal working of the NN system. Neural networks work on the *Black Box* principle. According to some publications such as [9], NN are the prediction models which have the biggest potential in predicting time series and high-frequency financial time series data.

In NN examples are brought forward the network and then the network tries to get as close as possible to the given output by adapting its parameters (weights). Neural network model has a large number of internal variables which are supposed to set up well in order to optimize the outputs.

In this section we firstly show an approach of function estimation for time series modelled by means of classic RBF trained by BP, and then trained by GA.

A. Classic RBF NN trained by PB algorithm

Because of not perfect classification ability as well as other cons of perceptron, RBF networks were developed [10]. RBF which stands for radial basis function is just like the perceptron, a feed-forward neural network. The architecture illustrated in Fig. 2 is quite similar to perceptron network, however there are some differences which include calculation of processing (hidden) neurons and different activation function of the output neuron.

The structure of a neural network is defined by its architecture (processing units and their interconnections, activation functions, methods of learning and so on).

Potentials of hidden perceptron (the first neural network of feedforward type) neurons are a simple linear operation of scalar product of vectors \mathbf{w} and \mathbf{x} . In RBF NN, the potential of the inner neuron is counted as follows: RBF NN defines potential of f th hidden neuron as a difference of Euclidian distance given by vectors

$$u^j = \|\mathbf{x} - \mathbf{w}^j\|^2, \text{ for } j = 1, 2, \dots, s \quad (12)$$

where s is the number of the RBF neurons. Note that for the RBF NN, the hidden layer weights \mathbf{w}^j represent the centers

e^j of activation function in the hidden layer. To find the weights or centers of activation functions we used the adaptive (learning) version of the K -means clustering algorithm [11].

RBF NN also uses different types of activation functions known as Gaussian or radial basis function. Activation function for f th hidden neuron is defined as

$$\psi_2(u^j) = e^{-\frac{u^j}{2\sigma_j^2}} = e^{-\frac{\|\mathbf{x} - \mathbf{w}^j\|^2}{2\sigma_j^2}} \quad (13)$$

for $j = 1, 2, \dots, s$, where σ_j^2 is a variance of f th neuron. If the components of input vector are not orthogonal then the activation function looks like this

$$\psi_2(u^j) = e^{-\frac{-(\mathbf{x} - \mathbf{w}^j)^T \Sigma^{-1} (\mathbf{x} - \mathbf{w}^j)}{2}} \quad (14)$$

The computation of the output can be described as

$$\hat{y} = \psi_3[\mathbf{v} \psi_2[\mathbf{w} \mathbf{x}_t]] \quad (15)$$

or in matrix form

$$\hat{y} = \psi_3 \left[\psi_2 \left[\begin{pmatrix} w_{11} & \dots & w_{1k} \\ \vdots & \ddots & \vdots \\ w_{s1} & \dots & w_{sk} \end{pmatrix} \times \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} \right] \times \begin{bmatrix} v_1 \\ \vdots \\ v_s \end{bmatrix} \right] \quad (16)$$

Activation function of output neuron is also different; output neuron is always activated by a linear function $y = x$.

The synaptic weights can be found by a simple learning rule known as the BP rule. In general, the network in Fig. 2 learnt so that the errors $e_t^{[3]}$ identified as $e_t^{[3]} = y_t - \hat{y}_t$ at the output node, where y_t is the desired output pattern or teacher, \hat{y}_t is the computed pattern, are propagated backwards and adapt the weights according to the following procedures:

- Compute the error for the output node

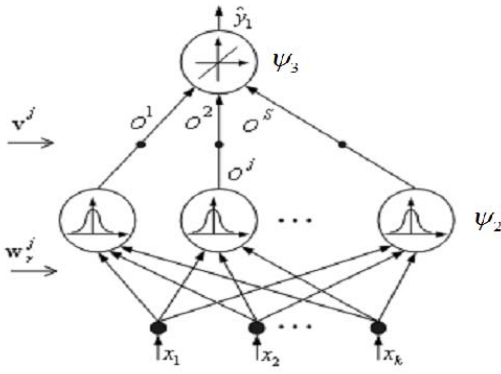


Fig. 2. RBF neural network architecture

$$e_i^{[3]} = (y_i - \hat{y}_i) \psi_3(u_j) \text{ for } j = 1, 2, \dots, s \quad (17)$$

- Update the connections v_j as

$$v_j^{new} = v_j^{old} + \eta o_j e_i^{[3]} \text{ for } j = 1, 2, \dots, s \quad (18)$$

where the term η , $\eta \in (0,1)$ is a constant called the learning rate, $e_i^{[3]}$ is the error given by $e_i^{[3]} = y_i - \hat{y}_i$.

- Compute the errors $e_j^{[2]}$ for the hidden layer nodes

$$e_j^{[2]} = e_i^{[3]} \psi_2'(u_j) v_j^{old}, \text{ for } j = 1, 2, \dots, s \quad (19)$$

- Update the connections w_{rj} as

$$w_{rj}^{new} = w_{rj}^{old} + e_j^{[2]} x_r \text{ for } j = 1, 2, \dots, s; r = 1, 2, \dots, k \quad (20)$$

Typically, the updating process is divided into epochs. Each epoch involves updating all the weights for all the examples.

B. RBF NN trained by GA

The weights $v_{j,i}$ can be adapted by genetic algorithms (GA) as well [12]. Genetic algorithms (see Fig. 3) are implemented as a computer simulation in which a population of abstract representations (called chromosomes) of candidate solutions (called individuals) to an optimization problem evolves toward better solutions.

The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on the fitness), and modify it (recombined and mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum a number of generations has been produced or a satisfactory fitness level has been reached for the population.

In the first two blocks of GA we define the initial population of neural network weights, optimization criteria, and fitness functions. Fitness function is set as minimization of the RMSE.

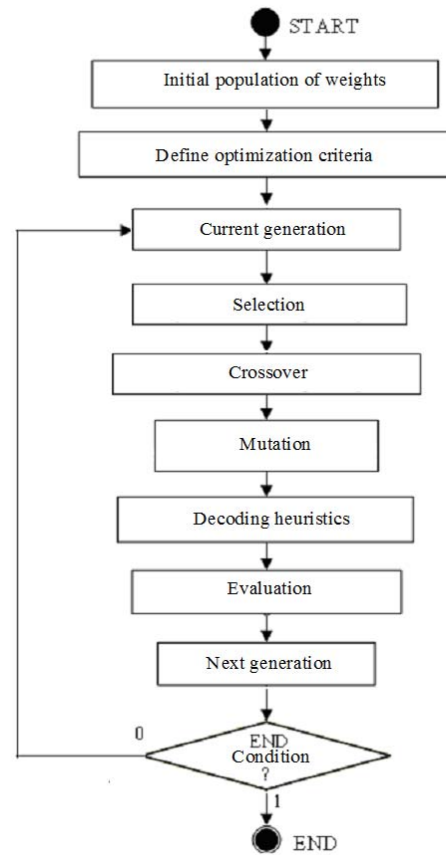


Fig. 3. Flow chart of common GA method

Genetic algorithms traditionally work with genes either 0 or 1. The initial population of weights \mathbf{v} was generated randomly from the interval $(a, b) \equiv (-0.7, 0.7)$ and transformed into the integer digit denoted as l by the following formula

$$l = [(v - a) / (v - b)](2^k - 1) \quad (21)$$

where is the value of weights (\mathbf{v}) randomly chosen from the interval (a, b) , k is the length of binary string in this case of size 16.

According to Darwin's evolution theory the best chromosomes from the population are selected to be parents to crossover. There are many methods how to select the best chromosomes, see e.g. [13, 14]. In this paper the roulette wheel selection was used. Mathematical foundation for the roulette wheel selection can be found in [15].

After the selection of two chromosomes follow two basic operators of genetic algorithm: crossover and mutation. In this work the single-point crossover has been applied. In the chromosome was randomly selected point which divide chromosome into two parts. Then those two parts of chromosomes were exchanged. After a crossover is performed, mutation take place. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Crossover and Mutation can then be as shown in Fig. 4(a) and Fig. 4(b) respectively. More information about crossover and mutation operators can be find, e.g., in [16].

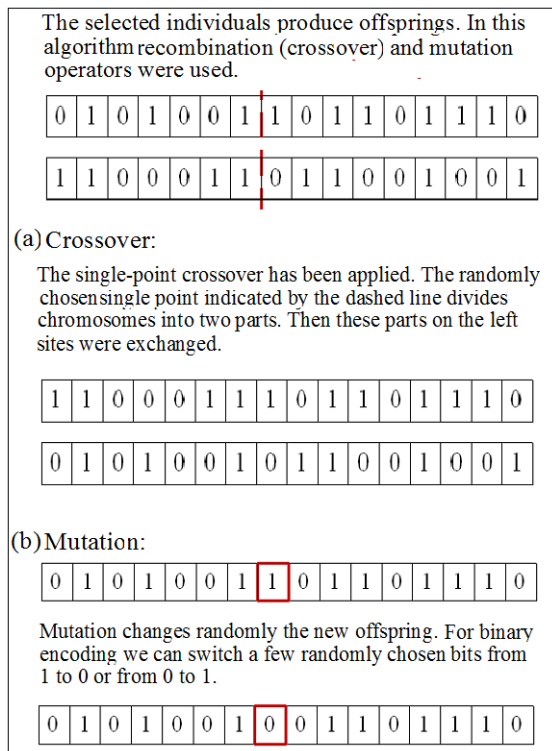


Fig. 4. Crossover (a) and Mutation (b) operators

With mutation is the process of creating a new individual completed. After decoding values of individuals and achieving the test conditions is algorithm stopped.

V. RESULTS AND EMPIRICAL COMPARISON

The parameter settings we used and the details of network topology and learning parameters to estimate the weights for RBF NN classic (BP learning algorithm used) and RBF NN with GA learning algorithm are given in Table I.

TABLE I. PARAMETER VALUES USED IN THE RBF NN TRAINED BY GA APPROACH

Model:	RBF NN classic	RBF NN with GA learning
Training/Testing Data	2928/1488	2928/1488
Initial value of v_j	0.01	(-0.7; 0.7)
Learning coefficient	0.01	0.01
Number of RBF neurons	92	65
Number of epochs	673	1528

There is no systematic method to determine these parameters in Table I. The optimum number of hidden layer nodes was found to be 92 and in hybrid model to be 65 as shown in Fig. 5(a) and Fig. 5(b) respectively. Training rate was 0.01. The MAE and MAPE functions for testing data set were calculated by the equations (6) and (7) respectively. Fig. 6(a) illustrates the convergence of MAE (error) function versus the number of training epochs for classic RBF NN, and Fig. 6(b) illustrates the convergence of MAE function versus the number of training epochs for RBF NN with GA learning algorithm GA respectively.

Both RBF NN approaches were trained using the variables and data sets as the statistical ARIMA(6,1,1)(1,1,1)₄₈ above.

In Table II are the accuracy results of the ARIMA, RBF NN and SVR methods expressed in term of MAE and MAPE.

MAE is a standard metric used by the research community and MAPE is the metric preferred by the industry forecasters. SVR is the most accurate method. (MAPE = 0.54), followed by ARIMA (MAPE = 1.32), RBF NN trained by GA (MAPE = 3.19%) and classic RBF NN trained by BP (MAPE = 3.5%). All proposed forecast models based on advanced statistical and soft computing methods have MAPE measures much less than 5%, i.e. they indicate that all forecast models are very good.

TABLE II. THE PERFORMANCE COMPARISONS FOR RBF NN, ARIMA(6,1,1)(1,1,1)₄₈ AND SVR APPROACHES

Approach	MAE	MAPE [%]
Classic RBF NN (BP learning)	323.294	3.50
RBF NN (GA learning)	301.088	3.19
ARIMA(6,1,1)(1,1,1) ₄₈	115.27	1.32
SVR	50.28	0.54

The use of SVR and ARIMA models is a powerful approach to the solution of many forecasting problems. But, they are not without several limitations. In both AIRMA and SVR models, there is not conventional way to modify or update the estimates of the model parameters as each new observation becomes available. In contrast to NN, another drawback of ARIMA models is, that there is the learning speed very slow. The estimate of the parameters can be not parallelized.

VI. CONCLUSION

In this paper, we considered predicting the electricity demand for half-hourly data. We adapted and evaluated machine learning methods successfully used prediction and compared their performance with the state-of-the-art statistical methods used for electricity demand prediction. We showed that SVR and ARIMA models outperformed the neural network models. Although we cannot generally to say that statistical models generally outperform NN models, we can say that NN models have equivalent prediction performance comparing to statistical models. We could also see, the NN have such attributes as computational efficiency, simplicity, and easy adjusting to changes in the process being forecast. The importance of having good intelligent forecasting tools for time series is ever more important with increasing number of data when more effort must be devoted to development of efficient data handling.

Future work will include exploring other ways of combining the prediction methods. Our main research objective is also to apply the developed meta-heuristic on various datasets or different time horizons. Selected metaheuristics will be tested with different parameter combinations, and the combination of parameters which can yield approximate feasible solution in an acceptable computation time.

ACKNOWLEDGMENT

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project „IT4 Innovations excellence in science - LQ1602.

REFERENCES

- [1] J. W. Taylor, "Short-term Electricity Demand Forecasting using Double Seasonal Exponential Smoothing," *Journal of the Operational Research Society*, Vol. 54, pp. 799-805, 2003.

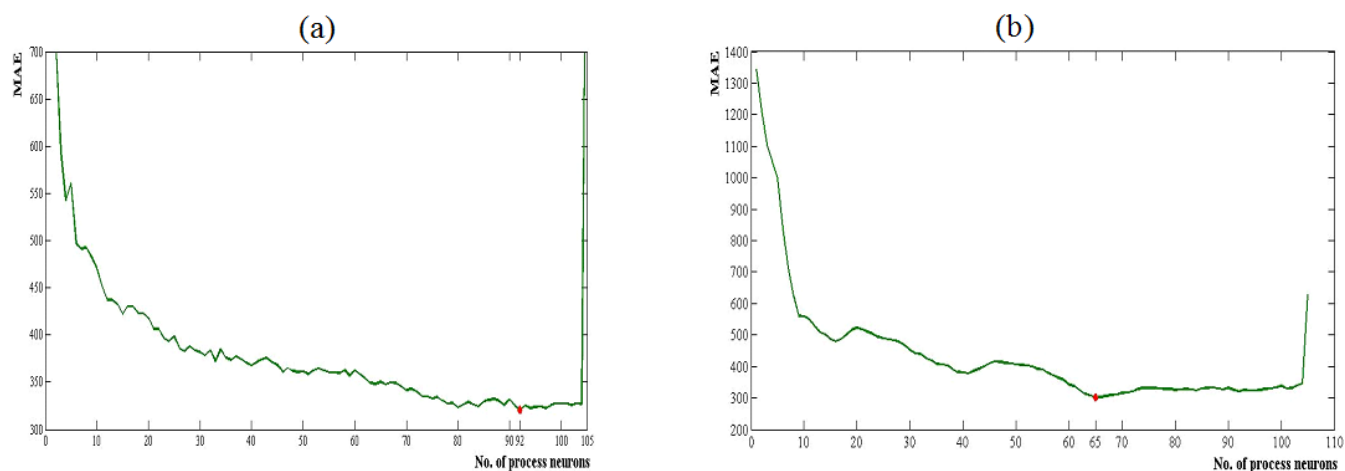


Fig. 5. Convergence MAE (error) function depending on the number of processing neurons for classic RBF NN (a). Convergence MAE function depending on the number of processing neurons for RBF NN + GA approach (b)

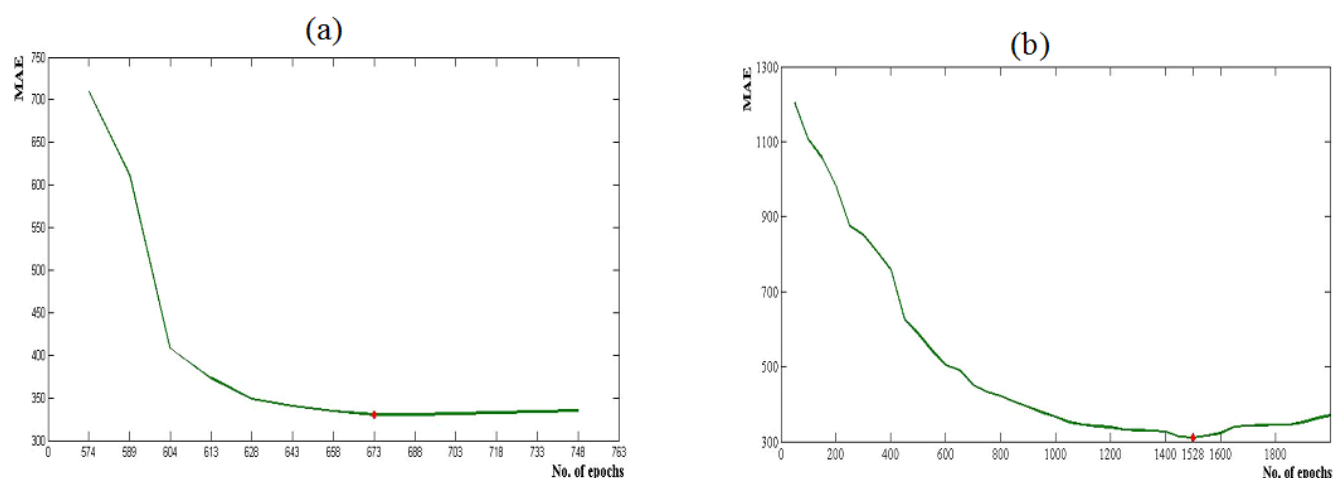


Fig. 6. Convergence MAE (error) function depending on the number of trained epochs for classic RBF NN (a). Convergence MAE function depending on the number of trained epochs for RBF NN + GA approach (b)

- [2] G. P. Box, and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, San Francisco: Holden-Day, 1976.
- [3] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic*, Massachusetts: The MIT Press, 2001.
- [4] J. A. K. Suykens, "Least Squares Support Vector Machines for Classification and Nonlinear Modelling," *Neural Network World* 1-2, pp. 29-47, 2000.
- [5] V. Vapnik, "The Support Vector Method of Function in Nonlinear Modeling: Advanced Black-Box Techniques," Suykens, J.A.K., Vondewalle, J. (Eds.), Kluwer Academic Publishers, Boston, pp. 55-85, 1998.
- [6] C. W. Therien, "Discrete Random Signals and Statistical Signal Processing," Prentice-Hall, USA, 1992.
- [7] D. C. Montgomery, A. J. Lynwood, J. S. Gardiner, *Forecasting and Time Series Analysis*, New York: McGraw-Hill, Inc., 1990, pp. 281-283.
- [8] M. Hall, E. Frank, G. Holmes, "The WEKA Data Mining Software," 2009, [Online] (3.6.6), Available at: www.weka.com, [Accessed november 2011].
- [9] J. G. Gooijer, R. J. Hyndman, "25 Years of Time Series Forecasting," *International Journal of Forecasting*, Vol. 22, pp. 443-473, 2006.
- [10] M. J. L. Orr, *Introduction to Radial Basis Function Networks*, Centre for Cognitive Science, University of Edinburgh, Scotland, April 1996.
- [11] D. Marcek, M. Marcek, J. Babel, "Granular RBF NN Approach and Statistical Methods Applied to Modelling and Forecasting High Frequency Data," *IJCIS*, Vol. 2(4), pp. 353-364, December 2009.
- [12] A. M. Zamba M., "A Genetic Algorithm Approach for Solving Cutting Stock Problem in Lumber Cutting Industry," *Proc. 18th Int. Conf. on Soft Computing*, eds. R. Matousek, Czech Republic, University of Pardubice, Pardubice, pp. 70-75, 2012.
- [13] N. Saini, "Review of Selection Methods in Genetic Algorithms," *International Journal of Engineering and Computer Science*, Vol. 6, Issue 12, pp. 22261-22263, December 2017.
- [14] S. Anand, N. Afreen, and S. Yazdani, "A Novel and Efficient Selection Method in Genetic Algorithm," *International Journal of Computer Application*, Vol. 129, No. 15, pp. 7-12, November 2015.
- [15] M. Obitko, "Genetic Algorithm," 1998, [Online], Available at: <http://www.obitko.com/tutorials>, [Accessed January 2012].
- [16] DEB Kalyanmoy, *Multi objective Optimizationary using Evolutionary Algorithms*, Wiley India Edition, 2005.

Classification of malicious emails

1st Eva Marková2nd Tomáš Bajtoš3rd Pavol Sokol4th Terézia Mézešová*Computer science institute**Computer science institute**Computer science institute**Computer science institute**Pavol Jozef Safarik University Pavol Jozef Safarik University Pavol Jozef Safarik University Pavol Jozef Safarik University*

Košice, Slovakia

Košice, Slovakia

Košice, Slovakia

Košice, Slovakia

eva.markova@student.upjs.sk tomas.bajtos@student.upjs.sk

pavol.sokol@upjs.sk

terezia.mezesova@upjs.sk

Abstract—An inherent part of everyday life and work on a computer is ownership and use of an email address. The main aim of this paper is to analyze existing approaches to classification of malicious emails. We have implemented a system, which is able to distinguish between legitimate and malicious emails. Subsequently, malicious emails are classified into three subcategories: spam, scam, and phishing. We prepared a labeled dataset. We extracted several features from emails contained in the dataset. Within the system, we have implemented four supervised machine learning methods (Random Forest, Decision Tree, Support Vector Machines, k-Nearest Neighbors) and evaluated them. According to our results, the Random Forest is the most suitable approach for email classification.

Index Terms—malicious email, phishing, spam, machine learning

I. INTRODUCTION

Every year, attacks on the Internet become more and more sophisticated. Attackers often get credentials or bank details through malicious emails, which may contain fraudulent links. They also may require users to install some malicious software which can monitor their activity on the computer. Today's emails no longer seem to be written by a machine, so it's difficult to distinguish illegitimate emails from the legitimate ones.

This problem needs to be solved, so we can prevent e.g. data leaks in an organisation. It is also important to users not to deal with the loss of money or credentials. We also want to make it easier for IT staff, because they often have to deal with problems, which can be avoided by creating a system capable of detecting malicious email.

Before the system was built, we previously tested employees and students in our organization, which is around 10.000 people [1]. We sent them three different versions of phishing email. It turned out that approximately 5 percent of people (about 500 people) responded to this email and that's a big loss for such an organization as university.

In this paper we compared the performance of 4 different classifiers designed to categorize emails into categories and we evaluated them by performance metrics. We used Random Forest, Decision Tree (CART), Support Vector Machines and k-Nearest Neighbors. It turned out that the best results are

This research is funded by the VVGS projects under contract No. VVGS-PF-2019-1062, VEGA project under contract No. VEGA/A-1/0056/18, and Slovak Research and development agency (SRDA) project under contract No. APVV-17-0561.

achieved by Random Forest and in our system we used default implementation offered by the scikit-learn library [2] in Python. In our system we also can see the results of others algorithms, but the most important is Random Forest.

The problem with classification is how to clearly distinguish between spam, scam phishing and legitimate email. Spam is any irrelevant or unsolicited message, mostly advertisement that is sent over the Internet. Scam is a popular form of fraud in which an attacker convinces a victim to pay a certain amount of money and promises a greater reward. Phishing contains links to malicious websites that appear legitimate, or encourages to click on a link, or attempt to retrieve sensitive information. It is very difficult to find characteristics of emails that clearly distinguish these categories. For example, in scam it is needed to look at the text part of an email, because it often does not contain any suspicious links or any attachment. The main goal of this paper is the analysis of the techniques for detection malicious emails and also implementation of malicious emails prevention system. We state the following research sub-goals:

- 1) comparison of methods for detection of malicious emails, and
- 2) design and implementation of system for detection of malicious emails.

This paper is organized into six sections. Section II focuses on the review of published research related to methods to detect malicious emails and design and implementation of prevention system. Section III focuses on research methodology. Section IV describes design and implementation of proposed system. Section V contains results and discussion and last section contains conclusion and our suggestions for the future research.

II. STATE OF ART

There are several approaches to protect against malicious emails. For example Natural Language Processing (NLP), blacklists, data mining algorithms and others. In most existing solutions, for detection are used data mining algorithms, because they turned out to be the most effective. In Tab. I we can see a comparison of the related works. They mostly focused on comparison of few approaches to detect malicious emails and also on design and implementation of a system. We compared which features and methods authors used in their papers. As part of a more detailed analysis, we focused

on by what features authors categorized email. In most of the papers, they have focused on URL-based features, because malicious emails often contain fraudulent links or lead to a phishing site. These include, for example, the number of URLs, the maximum number of dots in a URL and so on. Other types of attributes are header-based, such as sender domain, send time, domain address existence etc, which we can find them in email header. Body-based features are known as the most important. They include for example whether an email contains JavaScript, an HTML form or an attachment etc. Text attributes are important to distinguish for example scam from other categories. It is necessary to focus on the presence or quantity of some important words.

TABLE I
COMPARISON OF RELATED WORKS

Features	Methods				
	NLP	SVM	RF	DT	Other ^a
Textual	[3]	[4], [5], [8], [9], [10]	[4], [6], [10], [7]	[4], [7], [6], [10]	[4], [7], [5], [8], [11]
URL-based	[3]	[4], [9]	[12], [4]	[4], [13], [14]	[4]
Body-based	-	[4], [9]	[4], [12]	[4], [13], [14]	[4]
Header-based	-	[9]	-	[13], [14]	-

^a Other methods - k-NN, neural networks, Naive Bayes methods etc.

SEAHound is a system developed for detection of malicious emails [3]. Authors used techniques of Natural Language Processing. In emails, they focused on 4 main features and whether the email contains urgency, a “bad“ question, a malicious URL link and a nice greeting. The accuracy of this system is 95%.

Yasin and Abuhasan [4] have built an intelligent classifier that is capable of detection phishing emails. The model extracts 16 features from emails that are given a value of importance. On this model they tested and then compared 5 well-known classification techniques including J48, Naive Bayes methods, Support Vector Machines, Multilayer Perceptron and Random Forest. The best results were achieved with Random Forest.

Another system that was developed was called PFILFER [12]. The authors extracted 10 features like URL links or if the email contains JavaScript. They achieved 99,5% accuracy using the Random Forest algorithm.

Abu-Nimeh et al. [7] decided to compare the 6 most commonly used classifiers on phishing dataset. Classifiers include logistic regression, CART, SVM, RF and neural networks. They extracted 43 features from the emails and used 10-fold cross-validation. Random Forest achieved the best results and accuracy was 92,28%.

Form et al. [9] developed a system which handles with data which are URL-based and content-based and also monitors features that can change the behaviour of computer. They analyzed the email header to extract features such as message id, sender email and return path, because they can hardly be changed by attackers. The authors extracted 9 features from

emails. They used the Support Vector Machines to classify emails and achieved the accuracy of 97,75%.

Saberi et al. [11] used a set of classifiers to detect scam emails. This set combines the k-NN algorithm, Poisson distribution and Naive Bayes algorithms to improve system performance. The features used in this work are all textual. All classifiers were text-oriented with 2400 observed most frequently occurring terms, which made the system ineffective.

Toolan and Carthy in their research in [14] looked at the selection of features to detect spam and phishing. They extracted 40 features from each email, and they included information about the body of the email, the subject of the email, the URL links, the sender, and whether the email contains JavaScript. They evaluated the importance of each feature with the “Information Gain” and used the C5.0 Decision Tree as a classifier.

In [13] the authors focused on comparing data mining algorithms which was used in similar works and also testing the J48 algorithm. The results indicate that specific attributes need to be extracted during preparation phase.

In [5] authors compared performance of regularized neural network on three spam datasets with 4 machine learning algorithms. They focused on identifying spam email based on the content of the message.

Subasi et al. [6] used Decision Tree algorithms for spam email filtering. They compared performance of CART, C4.5, REP Tree, NB Tree, AD Tree, LAD Tree, RF and Rotation Forest (RoF) on spam email dataset. The best results gave RoF.

Authors in [6] proposed a Support Vector Machine based naive Bayes filtering system. This system constructs an optimal separating hyperplane, which divides samples into two categories. This algorithm is only applicable to textual emails.

In the next paper [8] authors focused on transformation of data before using classifiers of machine learning. Then they used SVM, RF and also C4.5 Decision Tree to classify emails as spam or non-spam. They found out that is effective to use distributed memory and distributed bag of words with cosine similarity and Autoencoder.

III. RESEARCH METHODOLOGY

In this chapter we are dealing with dataset and we describe its creation. The next part of this chapter is devoted to the most commonly used methods for the detection of malicious emails.

A. Dataset

For purpose of our research we used a dataset from the Enron-Spam project [15], which consists of 298 emails [16]. Subsequently we have categorized these emails. 150 emails are legitimate and the other 148 emails are malicious. We have also collected forged emails from people from the Cyber Security Laboratory at our university.

At the beginning we worked with a dataset that was divided by us into several categories. The problem was that there could be a high error rate due to various environmental factors

(bustle, fatigue, etc.), so we decided that it would be more appropriate if several people categorized these emails.

We created a form that contained several selected fake emails from the dataset. We asked people from the academic CSIRT team who had been taught in advance how to distinguish different categories of social engineering to help us categorize these emails. 63 emails were classified as spam, 25 were phishing, 4 were spear phishing, and 56 were scam. None of the emails was marked as hoax. That is why we decided to drop the hoax category as well as spear phishing, adding the 4 spear phishing emails we received to the phishing category. These results differ only slightly from how it was previously classified. At least we made sure that we sorted the emails correctly.

Above the dataset we compared 4 algorithms namely Random Forest, Support Vector Machines, k-Nearest Neighbors and Decision Tree, whose implementation was offered by scikit-learn library [2] in Python 3.7. After several tests, we determined that the training dataset would be 60% of the entire dataset, and the test dataset would be 40% of the dataset, so we tested as algorithms to classify 120 emails. A very important value is how many spurious emails the classifier has determined to be true positive (TP) and how many were mistaken and identified them as legitimate.

B. Methods

To achieve the best possible results, we decided to use classification algorithms.

Random Forest (RF) is a computationally efficient technique that can quickly work with large datasets. It brings very precise classification, a new way of determining variable meaning and the ability to model complex interactions between variables. It is very flexible in performing several types of statistical data analysis including regression, classification etc.

The principle of Support Vector Machines (SVM) [17] is the division of training data into two classes, but the division into more classes than two is not excluded. The goal is to learn the classification rule from the training set so it can be assigned a specific class to all new subjects in the future. This approach has been well developed for binary classification. It divides the area into two classes and determines which object belong to which class. More important is how the SVM works for $k > 2$, where k is the number of classes of dataset. One way to solve a problem is to divide it into a set of binary classification problems. The k -th classifier constructs a hyperplane between the last class and the $k-1$ other classes. Alternatively $\frac{k(k-1)}{2}$ hyperplanes may be constructed, separating each class from the other. These methods are called “One-against-the-Rest” and “One-against-One”.

The k-Nearest Neighbors (k-NN) algorithm is a non-parametric classification technique which is effective in statistical recognition applications [18]. This algorithm is one of the simplest and effective methods for classification. It calculates the distance between the test sample and all existing samples in the training set. It creates a decision based on its k -nearest samples after applying the weight function.

However, creating predictions with a small dataset also has disadvantages. The distance between each new email and all existing emails should be calculated before making decision, which is computationally difficult for a large number of emails.

In the structure of a **Decision Tree (DT)**, each inner node indicates a test on the attribute, each branch represents the result of the test data and each leaf node identifies a particular class. The highest node in the tree is the root node. There are various algorithms for creating a Decision Tree, such as CART, ID3, C4.5 etc. These are the greedy approaches in which the Decision Tree is constructed from top to bottom [19].

IV. DESIGN AND IMPLEMENTATION OF PROPOSED SYSTEM

The main aim of this paper was to design and implement system for detection and categorization of malicious emails. In this section we will describe the implemented system.

The proposed system is shown in Fig. 1 and consists of several parts, namely: (I) module for forwarding suspicious email to the server; (II) monitoring of incoming emails in the application – this includes checking the existence of a similar email; (III) extracting attributes from an email if it was not found a similar email in the database; (IV) the database where we store the extracted attributes and requests; (V) email classifier; (VI) admin interface and (VII) module to send user feedback.

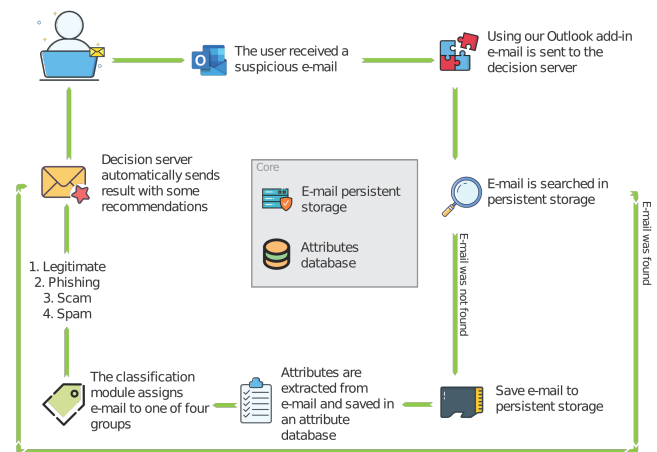


Fig. 1. Design of proposed system

A. Processing of emails

We have created an Add-in to Outlook, because in our university the employees are required to use this application to communicate with other employees. This Add-in was implemented in C# in Visual Studio 2017. It offers a template which directly creates a class that handles what should happen when application starts or shuts down. In our case, we have created a button which when pressed, it takes the email content including headers, body and attachments and forwards it as an attachment with an automatically generated subject and body content.

Within Outlook, we have implemented a script that checks whether a new email has been delivered to the currently logged in account. If an email that has been identified as new has the required subject, it controls if the attachment is similar to emails in our database.

To check if there is a similar email, we used the "fuzzy-wuzzy" library [20] in Python 3.7, which gave us the "ratio" function. This returns us the percentage of how much two emails are similar. In the implementation it uses the Levenshtein distance [21] to find similarity. After testing how the similarity value changes for some email changes, we set a threshold of 70%. If the similarity is lower, we cannot be sure if it is a similar email. Rather, we re-classify this email to make our results more accurate. If the similarity value is higher or equal than this threshold, we can safely say that there is a similar email in the database and therefore we can classify the email.

In our system, we decided to use the SQL database, containing 2 tables. The first table stores reports from users. On the other hand, the second table stores the analyzed requests.

B. Extracting attributes from email

If it is not possible to tell which category is email by similarity, it is necessary to extract the features from the email, which will allow us to classify the email. After studying the problem, we decided to omit header attributes. For example, it is no longer important to process for example sending time, because social engineering emails might be sent at different times.

In order to process and categorize emails, basic information, features extracted from URL links and the most commonly occurring words are important because for example scam does not normally contain links and by using only basic information might be difficult to identify scam.

Together we extract 11 attributes and also take into account the frequency of 35 specific words, which is crucial for us to classify into a particular category.

We decided to extract two basic information about the email that could help identify the fraudulent email, namely email size and number of attachments:

- **email size** - a very important parameter as it can tell us whether the email contains for example an attachment.
- **number of attachments** - the presence of an attachment can in combination with other parameters indicate that the email is dangerous.

Malicious emails often contain fraudulent links and it can be a good indication that they are malicious emails. We have looked at some of the features related to URL links:

- **number of URLs** - malicious emails often contain multiple URL links to fake websites.
- **maximum number of dots** - many attackers try to construct a legitimate-looking URL link by adding a few already known domains.
- **number of IP URLs** - some malicious websites may be hosted on computers that do not have DNS records.

- **number of image URLs** - these are images found in the "a href" tag.
- **number of URLs with @** - if the URL contains @, the web browser will ignore the part before this symbol and retrieve the address that follows after.
- **number of URLs with %** - an attacker can encode a part of a link with hexadecimal characters to hide a fraudulent URL.
- **number of URLs with a non-standard port** - we determine the number of URLs with a port different from 80 (http) and 443 (https).
- **maximum URL length** - attackers often use long URL links to hide the dubious part.
- **number of URLs** - that are different for the "a href" tag and the text after it - if there is a difference between the "a href" attribute and the text after it, we suspect it may be a malicious link.

Since we want to categorize malicious emails in our work, we also need to look at the frequency of words. The main reason is that malicious emails may not only contain URL links, so we would not be able to correctly classify for example scam.

The words we decided to extract are not random. Above each category in our prepared dataset, we launched a search for the most common words, with more than 10 occurrences and a word length of at least 5, to remove the words "the", "and" and so on. Since scam emails contain long texts, we decided to increase the occurrence parameter up to 25 for this category to avoid taking into account too many words. There are words like "information", "account", "transaction", "government" and so on.

C. Communication with client

Based on a comparison of algorithms, we decided to use the Random Forest algorithm to classify new records in our system, because it did the best result in several works. We have identified 4 different categories in which this algorithm can include a new email. These include spam, scam, phishing and legitimate emails.

In order not to neglect the human factor, the entire database must be available to the administrator, who would have the right to change the classifier's decisions. It may happen that the classifier makes the wrong decision and the administrator notices it, so it will be able to change this decision. The priority of the whole system is to automate it, but in extreme cases, user intervention is also necessary, as the technology is still not 100% correct.

The last module in this system is to send feedback to the user who provided the email. Because of this module, we also stored the user's email address in the database. The user receives an answer to their question if email is a legitimate or fraudulent and the steps they should take with respect to the identified category.

V. RESULTS AND DISCUSSION

In this section we describe results of comparison of the methods for detection and also results provided by our system.

A. Comparison of the methods for detection

First of all, it is important that our system correctly identifies legitimate and fake emails before classifying them into specific classes. So let's first see if we have chosen the appropriate attributes to identify a legitimate and malicious email. We use the performance metrics above to compare all values for each mentioned algorithm.

These algorithms categorized emails based on attributes email size, number of attachments, number of URLs, maximum number of dots, number of URLs in IP address, number of URLs in image, number of URLs with @, number of URLs with %, number of URLs the length of the URL, the number of URLs that are different for the "a href" tag and the text after it, as well as the presence of the most common words. For more details, see chapter 4.3 Extracting attributes from email.

In general, it is not important for us if the algorithm performs best in one case. After using 10-fold cross validation, we assessed that Random Forest could be the best option. In Tab. II, we can see the average success rate of each algorithm.

TABLE II
COMPARISON OF DATA MINING ALGORITHMS

	RF	k-NN	DT	SVM
True positive (TP)	0.94	0.80	0.91	0.91
True negative (TN)	0.89	1.00	0.87	0.87
False positive (FP)	0.11	0.00	0.13	0.13
False negative (FN)	0.06	0.20	0.09	0.09
Accuracy	0.91	1.00	0.89	0.87
Recall	0.94	0.80	0.91	0.91
Precision	0.92	0.89	0.89	0.89

In Tab. III, Tab. IV, Tab. V, and Tab. VI we can see how above mentioned algorithms (RF, k-NN, DT and SVM) classified emails. The columns show the expected classes of the emails and the row shows the predicted classes of the emails.

TABLE III
DECISION TREE (DT) - RESULTS

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	87.3%	1.8%	3.6%	7.3%
phishing	7.7%	38.5%	7.7%	46.1%
scam	16.7%	4.2%	74.9%	4.2%
spam	3.6%	7.1%	0.0%	89.3%

It can be noticed that each algorithm has a problem correctly classifying into the phishing category, with this class is the highest error rate. Obviously, it is not possible to determine exactly whether it is phishing. This may also be due to the fact that these algorithms had few phishing emails to learn from. It is important for us whether the email is classified as malicious and secondly if it falls into the category of phishing, scam or spam.

TABLE IV
RANDOM FOREST (RF) - RESULTS

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	89.2%	3.6%	3.6%	3.6%
phishing	7.7%	30.7%	7.7%	53.9%
scam	8.3%	0.0%	83.4%	8.3%
spam	3.6%	3.6%	0.0%	92.8%

TABLE V
SUPPORT VECTOR MACHINES (SVM) - RESULTS

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	87.2%	4.8%	3.2%	4.8%
phishing	20.0%	50.0%	20.0%	10.0%
scam	8.0%	0.0%	92.0%	0.0%
spam	4.5%	9.1%	4.5%	81.9%

It turned out that if we want our system to evaluate new emails as best as possible, it would be appropriate to use classification by Random Forest. It is also worth mentioning SVM, or there is a possibility to combine all algorithms.

B. Results for detection

It turned out that since the dataset is not balanced, SVM and k-NN do not classify emails correctly and tend to the majority class, so we decided to follow the results of the RF and DT (CART) algorithm.

If one of these algorithms is correct, we consider the email to be correctly classified. What matters is whether the algorithm correctly classified email as legitimate or malicious. In Tab. VII we can see the results of the comparison.

In this case, we can see that DT (CART) generally did better, but if we look at specific classes, we find that Random Forest is a little more accurate. Important value is true positive (TP), what is number of correctly classified malicious emails. It does not matter that the algorithm misclassified legitimate emails because the administrator can still change the system decision. In Tab. VIII and Tab. IX we can see how these algorithms classified emails.

The results of these algorithms are very similar. The RF algorithm incorrectly classified only 14% malicious emails and DT (CART) 24% emails. So we have confirmed that following result of the Random Forest algorithm is the right approach. We have created a system that can help many people at our university. Thanks to this system, administrators will be relieved of unnecessary work and will not have to solve problems with suspicious emails. This will be solved by our system, which if a similar email already is saved in the database, immediately classifies it into a specific class and automatically sends a reply to the user. It will help not only administrators, but also users who can verify any email that looks suspicious at any time.

VI. CONCLUSION AND FUTURE WORKS

According to the information available to us, there is no dataset with categorized emails into four groups like we

TABLE VI
K-NEAREST NEIGHBORS (K-NN) - RESULTS

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	100.0%	0.0%	0.0%	0.0%
phishing	23.0%	7.7%	0.0%	69.3%
scam	20.8%	0.0%	75.0%	4.2%
spam	17.8%	0.0%	0.0%	82.2%

TABLE VII
COMPARISON OF DATA MINING ALGORITHMS IN THE PROPOSED SYSTEM

	RF	DT (CART)
True positive (TP)	0.86	0.73
True negative (TN)	0.14	1.00
False positive (FP)	0.86	0.00
False negative (FN)	0.14	0.27
Accuracy	0.76	1.00
Recall	0.86	0.73
Precision	0.69	0.79

did. We have created the dataset on which we have tested effectiveness of 4 data mining algorithms. We reached the best results with the Random Forest and the accuracy was 90%.

Within the system, we have created an Add-in to Outlook that allows a user to forward a suspicious email to a server, we also monitor new emails on that server and if there arrives new email, system checks if there is a similar email - if it exists, we know it immediately categorize and send feedback to the user, if not, we classify email using the Random Forest algorithm, remembering the results of other algorithms.

In the future works we would like to create plugins into another email servers like Gmail and also plugins for another operating systems like Linux or Mac OS X. It is also important to create dataset with more emails in different languages.

ACKNOWLEDGMENT

This research is funded by the VVGS projects under contract No. VVGS-PF-2019-1062, VEGA project under contract No. VEGA/A-1/0056/18, and Slovak Research and development agency (SRDA) project under contract No. APVV-17-0561.

REFERENCES

- [1] P. Sokol, M. Glova, T. Mezešová, and R. Hučková, "Lessons learned from phishing test," *IDIMT 2017: Digitalization in Management, Society and Economy - 25th Interdisciplinary Information Management Talks*, pp. 297–304, 2017.
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [3] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. IEEE, 2018, pp. 300–301.
- [4] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *arXiv preprint arXiv:1608.02196*, 2016.
- [5] A. Barushka and P. Hájek, "Spam filtering using regularized neural networks with rectified linear units," in *Conference of the Italian Association for Artificial Intelligence*. Springer, 2016, pp. 65–75.

TABLE VIII
RANDOM FOREST (RF) IN THE PROPOSED SYSTEM

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	3.4%	3.4%	0.0%	17.2%
phishing	0.0%	10.3%	0.0%	6.9%
scam	6.9%	6.9%	13.8%	10.3%
spam	6.9%	0.0%	0.0%	13.8%

TABLE IX
DECISION TREE (DT) IN THE PROPOSED SYSTEM

Expected classes	Predicted classes			
	legitimate	phishing	scam	spam
legitimate	24.1%	0.0%	0.0%	0.0%
phishing	0.0%	17.2%	0.0%	0.0%
scam	17.2%	3.4%	17.2%	0.0%
spam	6.9%	13.8%	0.0%	0.0%

- [6] A. Subasi, S. Alzahrani, A. Aljuhani, and M. Aljedani, "Comparison of decision tree algorithms for spam e-mail filtering," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2018, pp. 1–5.
- [7] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM, 2007, pp. 60–69.
- [8] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive bayes algorithm for spam filtering," in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2016, pp. 1–8.
- [9] L. M. Form, K. L. Chiew, W. K. Tiong *et al.*, "Phishing email detection technique by using hybrid features," in *2015 9th International Conference on IT in Asia (CITA)*. IEEE, 2015, pp. 1–5.
- [10] M. Diale, T. Celik, and C. Van Der Walt, "Unsupervised feature learning for spam email filtering," *Computers & Electrical Engineering*, vol. 74, pp. 89–104, 2019.
- [11] A. Saberi, M. Vahidi, and B. M. Bidgoli, "Learn to detect phishing scams using learning and ensemble? methods," in *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*. IEEE Computer Society, 2007, pp. 311–314.
- [12] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," *CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE*, Tech. Rep., 2006.
- [13] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. Hossain, "Detection of phishing emails using data mining algorithms," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, 2015, pp. 1–8.
- [14] F. Toolan and J. Carthy, "Feature selection for spam and phishing detection," in *2010 eCrime Researchers Summit*. IEEE, 2010, pp. 1–12.
- [15] V. Metsis, I. Androustopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?" in *CEAS*, vol. 17. Mountain View, CA, 2006, pp. 28–69.
- [16] V. Metsis. (2006) Enron-spam dataset. [Online]. Available: <http://www2.aueb.gr/users/ion/data/enron-spam/>
- [17] J. Weston and C. Watkins, "Multi-class support vector machines," *Citeseer*, Tech. Rep., 1998.
- [18] H. Gascon, S. Ullrich, B. Stritter, and K. Rieck, "Reading between the lines: content-agnostic detection of spear-phishing emails," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 69–91.
- [19] A. K. Sharma, S. K. Prajapat, and M. Aslam, "A comparative study between naïve bayes and neural network (mlp) classifier for spam email detection," *Int. J. Comput. Appl.*, 2014.
- [20] SeatGeek. (2019) Project fuzzywuzzy. [Online]. Available: <https://github.com/seatgeek/fuzzywuzzy>
- [21] K. Beijering, C. Gooskens, and W. Heeringa, "Predicting intelligibility and perceived linguistic distance by means of the levenshtein algorithm," *Linguistics in the Netherlands*, vol. 25, no. 1, pp. 13–24, 2008.

Design of Microstrip Patch Antennas Operating on 2.45GHz in HFSS and FEKO

*Michal Márton, *Ľuboš Ovseník, *Ján Turán

*Department of Electronics and Multimedia
Communications of University of Technology Košice
Košice, Slovakia
michal.marton@tuke.sk
lubos.ovsenik@tuke.sk, jan.turan@tuke.sk

*Tomáš Huszaník, **Jakub Urbanský,
**Department of Electrical Engineering
of University of Technology Košice, Košice, Slovakia
michal.spes@tuke.sk, tomas.huszanik@tuke.sk

Abstract— Recent years are significantly affected by expansion of online services and their impact on human lives. People are inadvertently building addiction to technologies. Their lives shown different levels of addictions. With these problems are joined unwanted impacts. They need a lot of devices which each of them is able communicate. On the other hand of signals availability is the electromagnetic smog very dangerous. We are surrounding by a lot of radiation sources. These are build continually with regards on networks availability and high speed connections. 5G networks combine the many of technologies, communication protocols and standards. This paper is focused on design of microstrip patch antennas in two different professional software which are proposed for analysis of electromagnetic compatibility.

Keywords—Ansoft HFSS, FEKO, microstrip antenna

I. INTRODUCTION

We are living in the ages of expansion of communication systems supported by inventions, processing and upgrading of well-known approaches. These implementations allow expansions of communication networks. Each communication system needs input-output interface. Different systems have different interface characteristics. For optical networks which are able reach transmission rates about some of Gbps are these interfaces represented by laser (transmitter) and photodiode (receiver). Radio frequency (RF) communication use antennas on both sides. We are focused on design of microstrip patch antennas. Improving of their features is able due to modification of shape of conduct patch of microstrip patch antenna. These antennas are operating on frequency from non-licensed frequency range for Wi-Fi service [1-5].

II. DESIGN OF REFERENCE PATCH ANTENNA

On the base of theoretical knowledge was designed reference microstrip patch antenna which is used as base for next optimization process. The width of patch is defined by equation (1).

$$Wp = \frac{c}{2 \cdot f_0 \sqrt{\frac{\epsilon_r + 1}{2}}} \quad (1)$$

For determination of length of substrate we have to define effective value of dielectric constant ϵ_{reff} in (2)

$$\epsilon_{reff} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + 12 \frac{h}{W} \right)^{-1/2}, \quad (2)$$

then the effective length L_{eff} is defined as

$$L_{eff} = \frac{c}{2 f_0 \sqrt{\epsilon_{reff}}}. \quad (3)$$

Using effective length from (3) could be defined value length of edging ΔL

$$\Delta L = 0,412h \frac{(\epsilon_{reff} + 0,3) \left(\frac{W}{h} + 0,264 \right)}{(\epsilon_{reff} - 0,258) \left(\frac{W}{h} - 0,8 \right)}, \quad (4)$$

where h represents height of substrate. Finally, the length of patch is defined by (5).

$$Lp = L_{eff} - 2\Delta L. \quad (5)$$

On the base on equations (1-5) were obtained of reference microstrip patch antenna. These parameters are defined in Table I. Material of dielectric substrate is from Rogers Company denote as RO4350 with smaller value of dielectric constant in comparison with standard PCB (Printed Circuit Board) material FR04 [6-9].

TABLE I. PARAMETERS OF DESIGNED MICROSTRIP PATCH ANTENNA

Parameter	Describe
Frequency	2.45GHz
Substrate	Rogers RO4350
Relative permittivity	$\epsilon_r = 3.66$
Dimension of substrate	100x90x6mm
Dimension of patch	40x30x0.07mm

Dimensions of substrate are in format “W x L x H” (Width x Length x Height) and dimensions of patch are in format “Wp x Lp x Hp” (Width x Length x Height).

Dimensions of designed reference microstrip patch antenna are in the Fig. 1. Input port is used SMA connector which

connect two conduct layer placed on both sides of substrate [10-13].

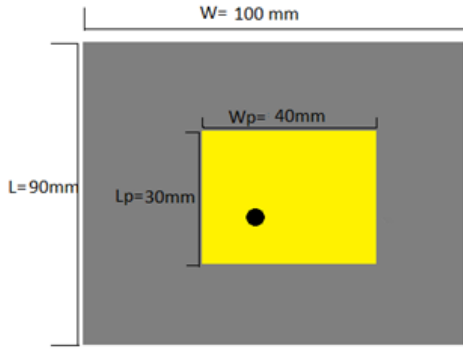


Fig. 1 Dimensions of microstrip patch antenna

In the Fig. 2 could be seen the designed model in development environment of Ansys HFSS software. The boundaries of antenna are represented by semi-transparent block.

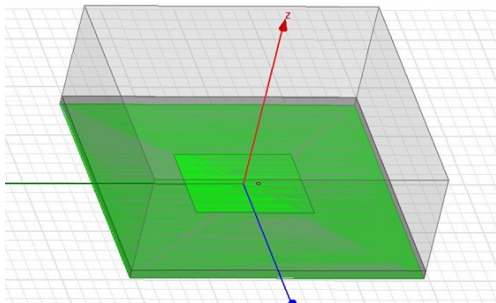


Fig. 2 Designed reference antenna in HFSS Ansys

The main parameter which represents impedance matching and resonance on operating frequency is $S_{1,1}$ parameter. $S_{1,1}$ parameter represents amount of radiated power on operating frequency. This parameter obtained through simulation of reference microstrip patch antenna without cut out is in the Fig. 3.

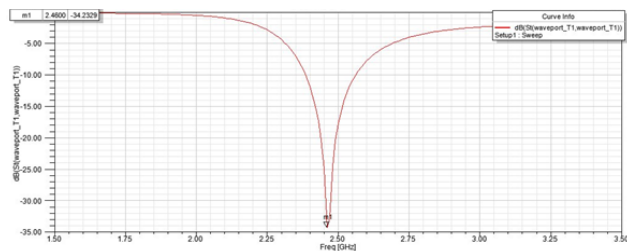


Fig. 3 $S_{1,1}$ parameter of designed reference antenna in Ansys

The $S_{1,1}$ parameter from Fig. 3 could be determine value of these parameter which correspond with -34,2329dB on operating frequency 2.45GHz.

Total radiation pattern of designed reference antenna placed in 3D shown on operating frequency 2.45GHz could be seen in the Fig. 4. The area with high level of saturation represents the area of maximal radiation.

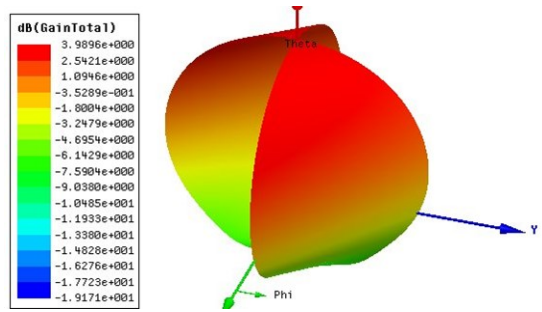


Fig. 4 Radiation pattern of designed reference patch antenna in HFSS Ansys

The radiation pattern of reference antenna in E plane cut which represents cut of electrical plane could be seen in the Fig. 5.

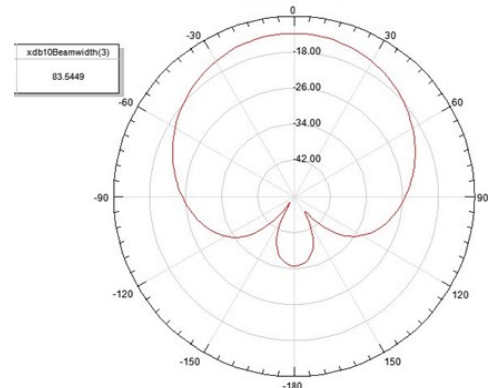


Fig. 5 Angle of radiation in polar coordinates realized for E plane cut

The radiation pattern of reference antenna in H plane cut which represents cut of electrical plane could be seen in the Fig. 6.

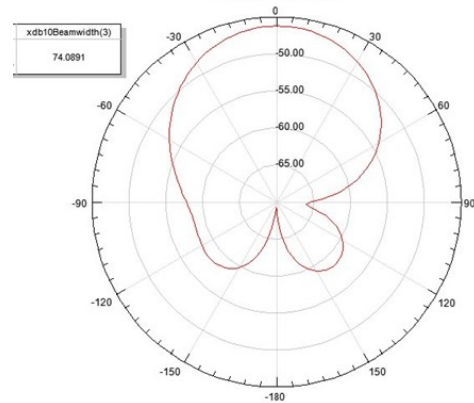


Fig. 6 Angle of radiation in polar coordinates realized for H plane cut

The parameter which represent maximum value of radiation is radiated power. The obtained value of radiated power from simulation of reference microstrip patch antenna is 1.003mW on operating frequency 2.45GHz.

III. MICROSTRIP PATCH ANTENNA WITH E-SHAPE CUT

In this section is realized design of cut out of patch. The dimensions of patch were obtained through optimisation process. These parameters of modified microstrip patch antenna with regards on ensure same operating frequency as reference antenna are in the Table II. [14-17].

TABLE II. PARAMETERS OF DESIGNED MICROSTRIP PATCH ANTENNA WITH E-CUT

Parameter	Describe
Frequency	2.45GHz
Substrate	Rogers RO4350
Relative permittivity	$\epsilon_r = 3.66$
Dimension of substrate	100x90x6mm
Dimension of patch	40x30x0.07mm

Dimensions of substrate are in format “W x L x H” (Width x Length x Height) and dimensions of patch are in format “Wp x Lp x Hp” (Width x Length x Height).

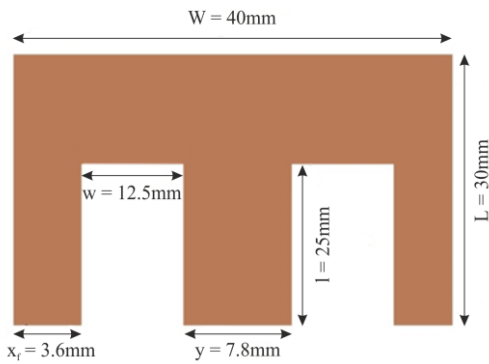


Fig. 7 Dimensions of E-shape cut out of radiation plane of patch

In the Fig. 7 could be seen parameters of patch. The cut out is E shape for increasing the value of gain and expand the radiation angle for ensure the cover of widely area in comparison with reference microstrip patch antenna. In the Fig. 8 could be seen the designed model in development environment of Ansys HFSS software. The boundaries of antenna are represented by semi-transparent block. The patch is placed on dielectric substrate and under substrate on the bottom layer is conduct grounding layer [18-20].

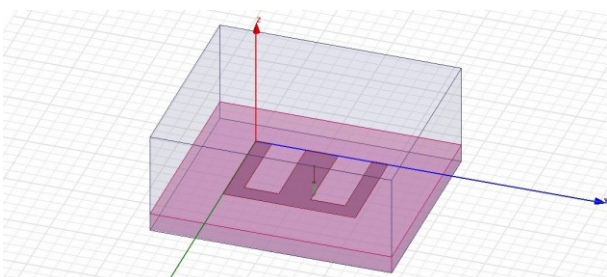


Fig. 8 Design of microstrip patch antenna with E-shape cut out in HFSS Ansys

The $S_{1,1}$ parameter from Fig. 10 could be determine value of these parameter which correspond with -14.98dB on operating frequency 2.45GHz.

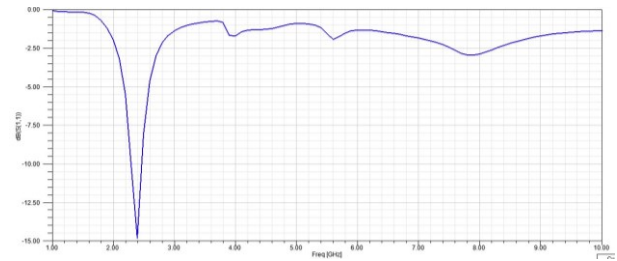


Fig. 9 S1,1 parameter of antenna with U-shape cut out in Ansys

Total radiation pattern of designed modified antenna placed in 3D shown on operating frequency 2.45GHz could be seen in the Fig. 10.

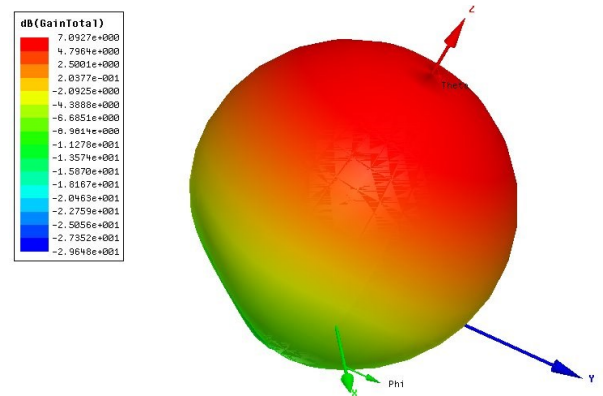


Fig. 10 Radiation pattern of patch antenna with E-shape cut out in HFSS Ansys

The radiation pattern of modified antenna in E plane cut which represents cut of electrical plane could be seen in the Fig. 11.

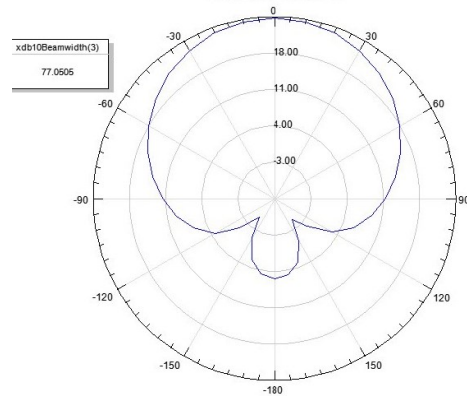


Fig. 11 Angle of radiation in polar coordinates realized for E plane cut out

The radiation pattern of modified antenna in H plane cut which represents cut of electrical plane could be seen in the Fig. 12.

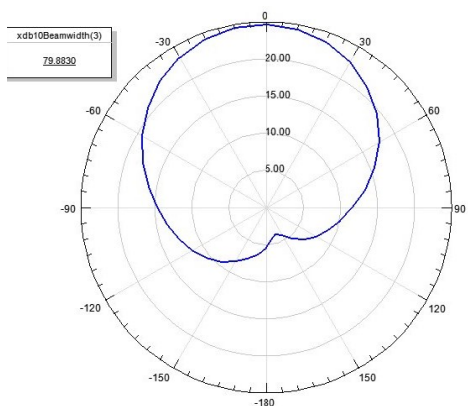


Fig. 12 Angle of radiation in polar coordinates realized for H plane cut out

The obtained value of radiated power from simulation of reference microstrip patch antenna is 2.634mW on operating frequency.

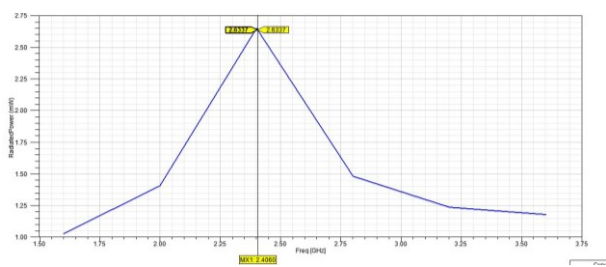


Fig. 13 Maximal radiated power of modified E-shape antenna

In the Fig. 13 could be seen distribution of radiation power which is maximal on operating frequency. The trend line show minimization of unwanted side radiation which can cause attenuation of other services which operating on frequencies nearby operating frequency of modified antenna.

In the Fig. 14 could be seen the designed same model in development environment of FEKO software. This design is proposed for result verification.

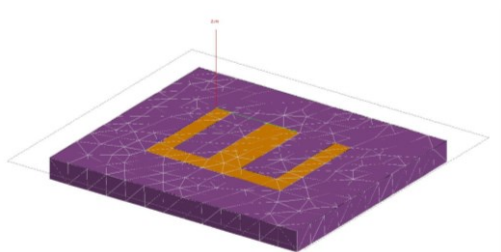


Fig. 14 Design of microstrip patch antenna with E-shape cut out in FEKO

The radiation pattern of designed microstrip patch antenna designed in FEKO with E-shape cut out is shown in the Fig. 15. The maximal obtained gain is given by value 6.7839dBi.

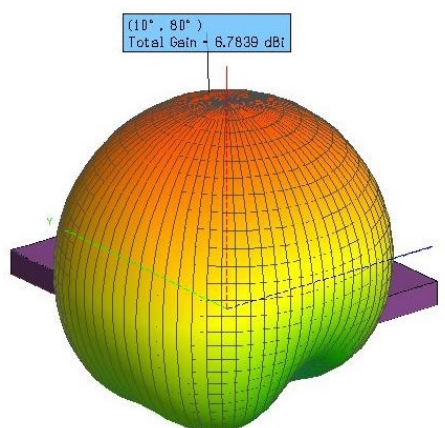


Fig. 15 Radiation pattern of patch antenna with E-shape cut out in FEKO

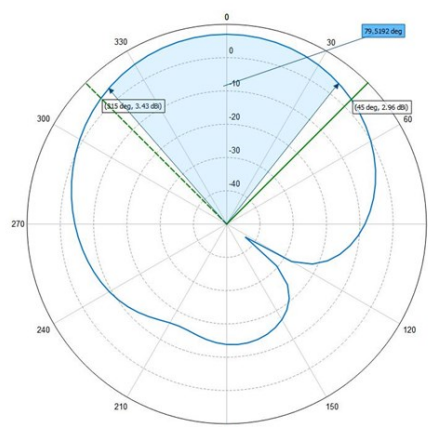


Fig. 16 Angle of radiation in polar coordinates realized for E plane cut out on half power of beam width

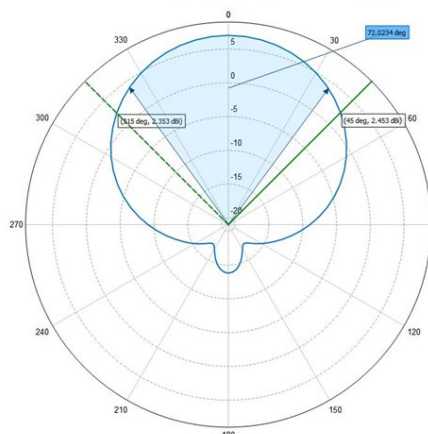


Fig. 17 Angle of radiation in polar coordinates realized for H plane cut out on half power of beam width

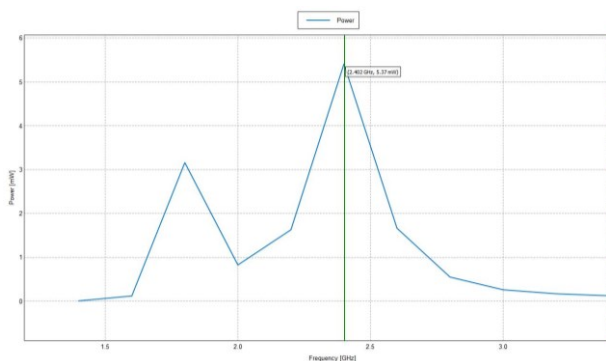


Fig. 18 Maximal radiated power of modified E-shape antenna in FEKO

IV. EXPERIMENTAL MODEL OF FSO SYSTEM

The comparison of these two antennas could be seen in the Table III. In this table is realized comparison of values of maximal radiated power, total gain, radiation angle in two different cuts.

TABLE III. COMPARISON OF REFERENCE PATCH ANTENNA WITH MODIFIED PATCH ANTENNA IN FEKO AND HFSS

Parameter	Reference antenna	Antenna designed in HFSS	Antenna designed in FEKO
Frequency	2.45GHz	2.45GHz	2.45GHz
Gain	3.99dBi	7.09dBi	6.78dBi
Radiation angle (horizontal cut)	74.09°	77.05°	72.02°
Radiation angle (vertical cut)	83.54°	98.47°	79.52°
Radiated power	1mW	2.61mW	5.37mW

V. CONCLUSION

The main comparison of these antennas is realized in the Table III. This comparison clearly shows increasing of parameters of antenna. Total gain of reference antenna obtained 3.99dBi and total gain of designed antenna in HFSS was 7.09dBi and total gain of designed antenna in HFSS was 6.78dBi. The impact of cut out in correct impedance matched shape dominantly affect the positive increasing of gain. The different between is significant in comparison with reference antenna and modified antenna in HFSS. The value of angle of radiation of reference antenna in E plane cut reach 74.09° and modified antenna in HFSS obtained 77.05° and modified antenna in HFSS obtained 72.02° The value of angle of radiation of reference antenna in H plane cut reach 83.54° and modified antenna in HFSS obtained 98.47° and modified antenna in HFSS obtained 14.93°. From the view of maximal radiated power obtain the highest-level antenna with E-shape cut out in FEKO.

ACKNOWLEDGMENT

This work was supported by following research grants: KEGA -023TUKE-4/2017 and APVV-17-0208 - Resilient mobile networks for content delivery.

REFERENCES

- [1] M. Márton, L. Ovseník, J. Turán, M. Špes, J. Vásárhelyi, "Possibility of increasing availability of FSO/RF hybrid system with implementation of helix antenna for 5.2GHz", 19th International Carpathian Control Conference (ICCC), 2018, 10.1109/carpathiancc.2018.8399681
- [2] J. Tóth, L. Ovseník, J. Turán, L. Michaeli, M. Márton, "Classification prediction analysis of RSSI parameter in hard switching process for FSO/RF systems", Measurement, vol. 116, pp. 602-610, 2018, 10.1016/j.measurement.2017.11.044
- [3] M. Márton, L. Ovseník, M. Špes, "Measurement effect of visibility in experimental FSO system" In: Informatics 2017. - Danvers: IEEE, p. 249-252. - ISBN 978-1-5386-0888-3, 2017
- [4] M. Márton, L. Ovseník, T. Huszanik, M. Špes, "Analysis of possibilities for measurement effect of visibility in experimental FSO system", Open Computer Science, vol. 8, no. 1, pp. 135-141, 2018, 10.1515/comp-2018-0013
- [5] M. Márton, L. Ovseník, J. Turán, M. Špes, J. Urbanský, "Comparison of microstrip patch antennas with different materials operating on 2.46GHz for FSO/RF hybrid system", 20th International Carpathian Control Conference (ICCC), 2019, 10.1109/CarpathianCC.2019.8765926
- [6] M. Márton, L. Ovseník, J. Turán, M. Špes, J. Urbanský, "Comparison of helix antennas operated on 2.4, 5.2 and 9.2GHz for FSO/RF hybrid system", 29th International Conference Radioelektronika (RADIOELEKTRONIKA 2019), 2019, 10.1109/RADIOELEK.2019.8733494
- [7] J. Tóth, L. Ovseník, J. Turán, "Free space optics experimental system - long term measurements and analysis", Acta Electrotechnica et Informatica. 2015. Vol. 15, no 2 (2015), pp.26-30
- [8] T. Ivaniga, P. Ivaniga, "Comparison of the optical amplifiers EDFA and SOA based on the BER and Q-factor in C-band", In: Advances in Optical Technologies, pp.1-9, 2017.
- [9] R. T. Cutshall, R. W. Ziolkowski, "Performance Characteristics of Planar and Three-Dimensional Versions of a Frequency-Agile Electrically Small Antenna", IEEE Antennas and Propagation Magazine, vol. 56, pp. 53-71, 2014, ISSN 1045-9243
- [10] P. Ivaniga, T. Ivaniga, "10 Gbps optical line using EDFA for long distance lines", Przeglad Elektrotechniczny, vol. 93, no. 3, pp. 193-196, 2017.
- [11] J. Volakis, Antenna engineering handbook, 1st ed. London: McGraw-Hill, 2007.
- [12] A. Deepak, P. Kannan, P. Shankar, "Design and Fabrication of Graphene Reinforced Polymer Conductive Patch-Based Inset Fed Microstrip Antenna", International Journal of Nanoscience, vol. 17, no. 0102, p. 1760019, 2018.
- [13] P. Ivaniga, T. Ivaniga, "Comparison of DPSK and RZ-DPSK Modulations in Optical Channel with Speed of 10 Gbps", vol. 41, no. 2, pp. 185-196, 2017.
- [14] A. Nag, K. Roy, D. Chaudhuri, "A Crown-Shaped Microstrip Patch Antenna for Wireless Communication Systems", Lecture Notes in Electrical Engineering, pp. 399-405, 2017.
- [15] G. Fedorko, P. Liptai, V. Molnár, "Proposal of the methodology for noise sources identification and analysis of continuous transport systems using an acoustic camera", Engineering Failure Analysis, vol. 83, pp. 30-46, 2018.
- [16] J. Zbojovsky, M. Pavlik, D. Medved, Z. Conka, L. Kruzela, "Shielding of the electromagnetic field in the range from 0.9 to 6 GHz by experimental shielding coatings", In: CANDO-EPE 2018, Budapest, Hungary, pp. 273-276, 2018
- [17] G. Chenhu et al., "A half-space covered antenna for air-ground communication", 2017 Sixth Asia-Pacific Conference on Antennas and Propagation (APCAP), 2017.
- [18] O. Kováč, J. Mihalík, I. Gladišová, "Convolution implementation with a novel approach of DGHM multiwavelet image transform," Journal of Electrical Engineering, vol.68, no.6, p.455-462, 2017.
- [19] IEEE Standard 211-1997, "The IEEE Standard Definitions of Terms for Radio Wave Propagation", December 1997, 27 p.
- [20] L. Blake, M. Long: Antennas, Raleigh, NC: SciTech Pub., 2009.

Seeking Relevant Information Sources

Marek Menšík

Department of Computer Science
VSB-Technical University of Ostrava
 Ostrava, Czech Republic
 mensikm@gmail.com

Marie Duží

Department of Computer Science
VSB-Technical University of Ostrava
 Ostrava, Czech Republic
 marie.duzi@gmail.com

Adam Albert

Department of Computer Science
VSB-Technical University of Ostrava
 Ostrava, Czech Republic
 adam.albert.st@vsb.cz

Vojtěch Patschka

Department of Computer Science
VSB-Technical University of Ostrava
 Ostrava, Czech Republic
 vojtech.patschka@vsb.cz

Miroslav Pajr

Institute of Computer Science
Silesian University in Opava
 Opava, Czech Republic
 mirapajr@seznam.cz

Abstract—In this paper we deal with the problem of seeking relevant information sources selected from scientific or other electronic publications. In the era of information surfeit, it is getting more and more difficult to extract relevant and reliable sources of information from the huge number of e-sources. The starting point is user's query for a given concept or topic. Our algorithm applies machine learning methods in order to propose hypothetic explications of the sought terms based on pieces of information extracted from the potentially relevant e-sources. Hypotheses, formalized in the TIL-Script language, are incrementally built by applying heuristic functions. The user thus obtains as closed approximations of the meaning of the sought terms as possible that at the same time provide fine-grained keyword definitions. As a result, it should be much easier to decide which of the e-sources are relevant for user's interest.

Keywords—*machine learning, TIL-Script, Information sources*

I. INTRODUCTION

With the increasing number of information resources that we have at our disposal, we encounter the problem of *information overload*. When looking for a particular piece of information, there are often several large resources to draw on yet only some of them are relevant for our search. In addition, another problem may crop up, namely that of comparing search results as for their compatibility, similarity, or even inconsistencies that may occur among them. Thus, we are confronted with the need to study carefully a large amount of different information resources in order to find a relevant one. All these efforts are time and power consuming, because we often have to go through many irrelevant resources until we encounter the right one. In this paper we propose a method promoting automatic information retrieval that would make it possible to select relevant information resources out of many irrelevant ones in an easier way.

To achieve the goal, we deal with logical natural language processing, because we want to perform the searches not only in the syntactic way, i.e. based on the specified key-words, but also in the semantic way by logically analyzing particular key-words. To this end we exploit the system of Transparent

Intensional Logic (TIL, see [1] and [10]), combined with the methods of supervised machine learning adjusted to natural language processing. TIL is an expressive higher-order logic with *procedural* semantics apt for the analysis of natural language. Similar ideas have been realized in “in press” [3] where we introduced the application to a multi-agent system; the agents learn to recognize geometric objects, specifically lancet arches. Then we concentrated on natural language processing; as a result, we applied an adjusted system of machine learning methods to the problem of refining and explicating simple concepts. For details, see “in press” [4].

In the effort to find explications of a simple concept, we search for the sentences that use the concept, analyse the sentences in TIL and gradually build up a compound concept that explicates the simple one. In “in press” [4] we perform the search only over one text document. In this paper we generalize the method so that to search over many text resources. In this way we obtain many different compound concepts that more or less explicate the sought pieces of information. The user thus can decide which of them is relevant. To make this decision easier, the selection of potential information resources is narrowed down by applying a post-processing semantic filter.

This research is interdisciplinary; in pursuit of a common goal, logic, computational linguistics and computer science jointly cooperate here. First, the text data are linguistically and logically processed so that TIL constructions are extracted by the Natural Language Logical Analyzer algorithm [2]. Then we transform these constructions into the TIL-Script language, which is a computational variant of TIL, to obtain input for the machine learning algorithms. Finally, the post-processing of the results is again the domain of logic. We compare the compound constructions obtained from different resources as for their consistency, compatibility and similarity to inform the user on potential discrepancies or even fake data.

The rest of the paper is organized as follows. Chapter II is a brief introduction into TIL. In Chapter III we summarise the principles of supervised machine-learning methods based on

symbolic representation. Inductive heuristics applied in these methods and their adjustments to natural language processing are introduced in Chapter IV. In Chapter V we describe the algorithm of searching and selecting relevant resources. Chapter VI illustrates the results by an example of explicating the concept of wild cat using information extracted from several documents. Concluding remarks can be found in Chapter VII.

II. FOUNDATIONS OF TIL

In TIL, expressions encode *algorithmically structured procedures* as their meanings. These procedures produce extensional or intensional entities, or even lower-order procedures, as their products. This approach has been informed by an *algorithmic turn* in semantics and advocated by, for instance, Moschovakis in [6]. Yet much earlier, in the early 1970s, Pavel Tichý defined six kinds of such meaning procedures that he coined TIL *constructions* as the centrepiece of his system, see [9] or [10].

The syntax of TIL is a hyperintensional, typed λ -calculus of partial functions. However, TIL λ -terms do not denote functions; rather they denote *procedures* (*constructions* in TIL terminology) that produce functions or functional values as their products. A linguistic sense of an expression is an abstract *procedure* detailing how to arrive at an object (if any) of a particular logical type denoted by the expression.

There are two kinds of TIL constructions, atomic and compound. *Atomic* constructions (*Variables* and *Trivializations*) do not contain any other constituent but itself; they supply objects (of any type) on which compound constructions operate. *Variables* x, y, p, q, \dots construct objects dependently on a valuation; they *v-construct*. *Trivialisation* of an object X (of any type, even a construction), in symbols $'X$, refers to or displays X without the mediation of any other construction. There are two dual compound constructions, namely *Composition* and *Closure*. *Composition* $[FA_1 \dots A_n]$ is the procedure of applying a function f (*v-constructed* by F) to a tuple argument A (*v-constructed* by $A_1 \dots A_n$). It *v-constructs* the value of f at A , if the function f is defined at A , otherwise the Composition is *v-improper*, i.e., it fails to *v-construct* anything. To produce a function rather than its value, we use *Closure* $[\lambda x_1 \dots x_n X]$. It is a procedure of *v-constructing* a function by abstracting over the values of variables x_1, \dots, x_n in the ordinary manner of λ -calculi. Finally, higher-order constructions producing lower-order constructions can be executed twice over. This is achieved by a fifth construction called *Double Execution*, 2X , that behaves as follows: If X *v-constructs* a construction Y , and Y *v-constructs* an entity Z , then 2X *v-constructs* Z ; otherwise 2X is *v-improper* by failing to produce anything.

TIL constructions, as well as the entities they construct, all receive a type. The formal ontology of TIL is bidimensional; one dimension is made up of constructions, the other dimension encompasses non-constructions. On the ground level of the type-hierarchy, there are non-constructional entities unstructured from the algorithmic point of view belonging to a *type of order 1*. Given a so-called *epistemic* (or '*objectual*')

base of atomic types (o -truth values, ι -individuals, τ -time moments/real numbers, ω -possible worlds), the induction rule for forming functions is applied: where $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \dots \times \beta_n$ to α , denoted $(\alpha \beta_1 \dots \beta_n)$, is a *type of order 1* as well. The above epistemic base $\{o, \iota, \tau, \omega\}$ was chosen, because it is apt for natural-language analysis, but the choice of base depends on the area to be analysed. Constructions that construct entities of a type of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted by $*_1$. This type $*_1$ together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, type $(\alpha \beta_1 \dots \beta_n)$, involving $*_1$ in their domain or range is a *type of order 2*. Constructions that construct entities belonging to a type of order 1 or 2 are *constructions of order 2*. They belong to a *type of order 3*, denoted $*_2$; any collection of partial mapping involving $*_2$ in their domain or range is a *type of order 3*. And so on *ad infinitum*.

III. SUPERVISED MACHINE LEARNING

In machine learning methods the supervisor teaches an agent by classified examples. All the examples are described by two types of attributes, namely input attributes and an output attribute. For example, meteorological situation can be described by input attributes such as *air pressure, air humidity, wind speed*, etc. The output attribute will then be the type of weather like *sunny, partly cloudy* or *rainy*. Hence, there is a functional dependency f between the values of input attributes and the output attribute. The aim of supervised machine learning is to learn this dependency by observing the values of input and output attributes in the training examples provided by the supervisor, and thus predict the output value given the input values. The learned function, called a *hypothesis*, approximates the original unknown function f as closely as possible. In case of the above meteorological data example, the agent aims to create a hypothesis predicting the type of weather. Correctness of the created hypothesis is verified by a set of test examples providing the values of input attributes. The learned hypothesis is correct if the agent correctly predicts the values of the output attribute. More details can be found in [5], [7], [8], [12].

In "in press" [3], the symbolic methods of supervised machine learning are discussed, and the chosen symbolism is the language of TIL constructions. The method has been applied to agents' learning and recognizing geometric figures. Adjustment of the method to learning explications of a simple concept by positive and negative examples extracted from natural language texts has been introduced in "in press" [4]. The role of an output attribute is now played by the simple concept to be learnt. Instead of input attributes there are constituents of molecular concepts that explicate the learnt concept. For example, to explicate the simple concept of cat, i.e. Trivialization $'Cat$ of the property of being a cat of type $(o\iota)_{\tau\omega}$, we can use as positive examples sentences describing this property, such as "*Cat is a predatory mammal that has*

been domesticated". Formalization in TIL comes down to this construction.

$$\lambda w \lambda t \lambda x \left[\left[[\textit{Predatory} \textit{Mammal}]_{wt} x \right] \wedge \left[\textit{Domesticated}_{wt} x \right] \right]$$

Types. *Predatory*/ $(ol)_{\tau\omega}$: property modifier; *Mammal*, *Domesticated*/ $(ol)_{\tau\omega}$: properties of individuals; $w \rightarrow \omega$, $t \rightarrow \tau$, $x \rightarrow \iota$: variables ranging over possible worlds, times and individuals, respectively. The whole Closure constructs the property of individuals, i.e. an object of type $(ol)_{\tau\omega}$.

The algorithm has created the above construction according to our method of analysis. Having assigned types to the objects the sentence "Cat is a predatory mammal that has been domesticated" mentions, formalization of the property of being a predatory mammal that has been domesticated is type driven as follows.

The property modifier *Predatory* is a function that takes as an argument a property to be modified, in this case the property of being a Mammal. Thus, the algorithm composes Trivializations of these two properties to obtain a modified property:

$$[\textit{Predatory} \textit{Mammal}] \rightarrow (ol)_{\tau\omega}$$

Since this property is to be applied to an individual produced by the individual variable x , it must be extensionalized first: $[[[\textit{Predatory} \textit{Mammal}]_w]t] \rightarrow (ol)$, or $[\textit{Predatory} \textit{Mammal}]_{wt}$ for short. Composing the result with the variable x ranging over individuals, a truth-value is produced:

$$[[[\textit{Predatory} \textit{Mammal}]_{wt} x] \rightarrow o$$

Similarly the property of being Domesticated is extensionalized and applied to the individual v -produced by x :

$$[\textit{Domesticated}_{wt} x] \rightarrow o$$

Composing these two constructions by conjunction, we obtain

$$[[[\textit{Predatory} \textit{Mammal}]_{wt} x] \wedge [\textit{Domesticated}_{wt} x]]$$

Finally, by abstracting over the values of free variables w , t and x the λ -Closure is obtained that produces our property, as it should be:

$$\lambda x [[[\textit{Predatory} \textit{Mammal}]_{wt} x] \wedge [\textit{Domesticated}_{wt} x]] \rightarrow (ol)$$

$$\lambda w \lambda t \lambda x [[[\textit{Predatory} \textit{Mammal}]_{wt} x] \wedge [\textit{Domesticated}_{wt} x]] \rightarrow (ol)_{\tau\omega}$$

By exploiting positive examples, we refine the hypothetic concept by inserting new constituents into it, or by generalizing its current constituents to prevent too special explication. For example, the positive example "Cat is a wild feline predatory mammal" triggers insertion of the property modifier *Feline* to the constituent $[\textit{Predatory} \textit{Mammal}]$ and generalization of the constituent $\textit{Domesticated}$. Indeed, cats are domesticated or wild. As a result, we have the hypothesis:

$$\lambda w \lambda t \lambda x \left[\left[[\textit{Feline} [\textit{Predatory} \textit{Mammal}]]_{wt} x \right] \wedge \left[[\textit{Domesticated}_{wt} x] \vee [\textit{Wild}_{wt} x] \right] \right]$$

By negative examples we specialize our hypothetic explication so as to refine it and distinguish from other similar concepts. For example, the negative example "Dog is a domesticated predatory mammal that barks" refines the concept of a cat by inserting properties *Bark* and *Dog* in negatives. The hypothesis is then specialized as follows:

$$\lambda w \lambda t \lambda x \left[\left[[\textit{Feline} [\textit{Predatory} \textit{Mammal}]]_{wt} x \right] \wedge \left[[\textit{Domesticated}_{wt} x] \vee [\textit{Wild}_{wt} x] \right] \wedge \left[\neg [\textit{Bark}_{wt} x] \wedge \neg [\textit{Dog}_{wt} x] \right] \right]$$

IV. METHODS OF SUPERVISED MACHINE LEARNING

The general framework of machine learning consists of the learning *objectives*, *training data* and *heuristic methods* for manipulating the symbolic representation of the data. In our case, the goal of learning is to find explications of a simple concept of an object O . In TIL, the simple concept is simply Trivialization of the object O , \textit{O} . Explications are closed molecular constructions producing an object O' that is as close to O as possible. Training data are natural language sentences in which O receives mention formalized in TIL language of constructions. We use three heuristic methods that work with a hypothetic explication. They are *Refinement*, *Generalization* and *Specialization*.

For our purpose we voted for an adjusted version of Patrick Winston algorithm [12] of supervised machine learning. This algorithm applies the principles of generalization and specialization to obtain a plausible hypothesis, i.e. a functional dependency between input and output attributes. As explained above, the adjustment concerns concepts, i.e. closed constructions, instead of attributes.

Another difference is the fact that in addition to generalization and specialization, we also use the method of *refinement*. By refining a hypothetic concept, we insert new constituents into the molecular construction learned so far.

Generalization usually consists in replacing one or more constituents of the hypothetic concept by a more general one, which is either extracted from agent's ontology or created from special constituents by composing them in the disjunctive way. As a special case, generalization can also be applied on numerical values of attributes. For instance, if we obtain a piece of information that the in-heat period of a wild cat is 2 days and another positive example specifies 8 days, we generalize it to the interval 2 – 8 days.

Specialization is triggered by negative examples. As a result, negation of a property that does not belong to the essence of the hypothetic concept is inserted. Specialization serves to distinguish the concept from similar ones. For instance, a wooden horse can serve as a negative example to the concept of horse, because a wooden horse is not a horse; rather, it is a toy horse though it may look like a genuine living horse.

Heuristic methods of the original Winston algorithm work with examples that cover all the attributes of a learned object. Based on positive examples the hypothesis is modified in such a way that the values of attributes are adjusted, or in case of a negative example an unwanted attribute marked as *Must-not-be* is inserted. In our application the sentences that mention the learned concept contain as constituents some but not all the requisites of this concept, and we build up a new molecular concept by adding new information extracted from positive or negative examples. Hence, we had to implement a new heuristic *Concept-introduction* for adding concepts of new requisites into a hypothetic concept. Negative examples trigger the method *Negative-concept* that inserts a concept of negated property into the hypothesis. Generalization is realized by modules that introduce a concept of a more general property; to this end we also adjusted the original heuristic *Close-interval* so that it is possible to generalize values of numeric concepts by the union of interval values.

V. DESCRIPTION OF THE ALGORITHM

Here is a brief specification of the algorithm.

A. Refinement

- 1) Compare the model hypothesis (to be refined) and the positive example to find a significant difference
- 2) If there is a significant difference, then
 - a) if the positive example contains as its constituent a concept that the model does not have, use the ***Concept-introduction***
 - b) else ignore example

B. Specialization

- 1) Compare the model hypothesis (to be refined) and the negative example to find a significant difference
- 2) If there is a significant difference, then
 - a) if the negative example has a constituent of the concept that the model does not have, use the ***Negative-concept***
 - b) else ignore example

C. Generalization

- 1) Compare the model hypothesis (to be refined) and the positive example to determine a difference
- 2) For each difference do
 - a) if the model and example differ at some constituents, then
 - i) if the constituents in which the model and example differ have the most specific general concept, use the ***General-concept***
 - ii) else use ***Disjunctive-concept***
 - b) if the model and example differ at an attribute numerical value or interval, use the ***Close-interval***
 - c) else ignore example

The whole process can be schematically illustrated by Fig. 1.

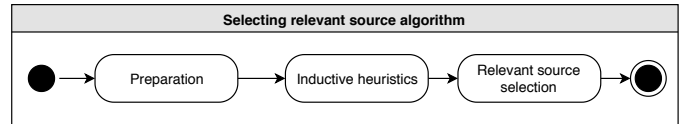


Fig. 1. Selecting relevant source algorithm

The *Preparation* module performs linguistic and logical analysis of the source texts. As a result, TIL constructions are produced. Next, we filter out constructions that do not mention the simple concept to be explicated. The set of so-obtained relevant constructions serves as an input for the section *Inductive heuristics*. This module produces hypothetic molecular concepts that should explicate the simple concept to be learned. Since there are several resource text documents from which the hypothetic concepts are extracted, we obtain several explications of the learned simple concept.

Fig. 2 is a schema of the Preparation phase.

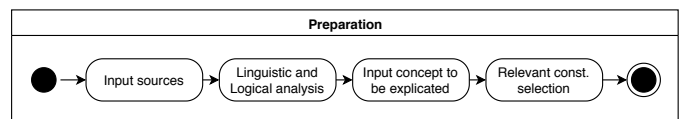


Fig. 2. Preparation

The inductive heuristic modules are schematically specified by the following Fig. 3.

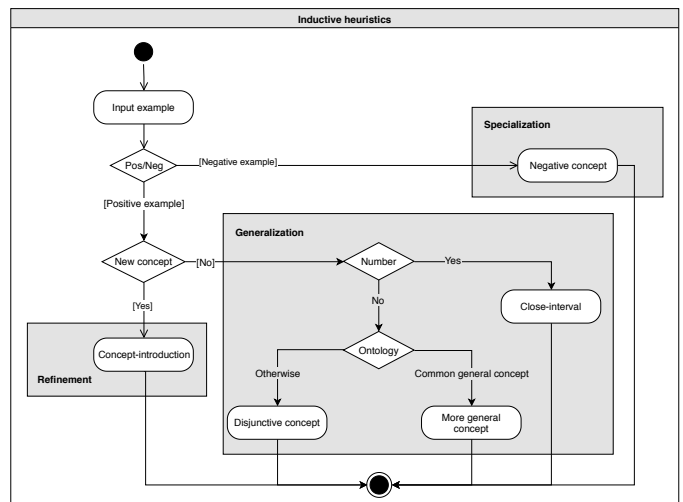


Fig. 3. Inductive heuristics

Since our algorithm is applied on several text resources, we obtain several molecular concepts that should explicate the simple one. The user thus can decide which of them are relevant and vote for the corresponding text documents. However, there still can be too many of them. Hence, to make the selection easier, there is the module *Relevant source selection* that performs post-processing of these explications. The module filters out irrelevant resources according to additional user defined criteria. Moreover, the system offers the

possibility of checking on *inconsistencies* or *contradictions*. For example, if we find contrary sentences like "All wild cats are mammals" and "No wild cat is a mammal", it is obvious that both cannot be true (though possibly both can be false). Now it is up to the user to decide which explication is correct. In addition, such incompatibilities can reveal texts that are deliberately misleading, incorrect, or these surprising incompatibilities can even be a sign of fake news.

Schematically, this last module of the algorithm is illustrated by Fig. 4.

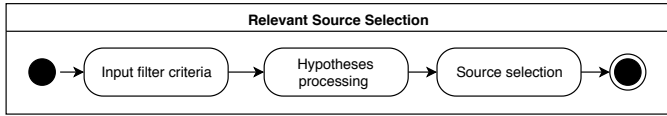


Fig. 4. Relevant source selection

VI. EXAMPLE OF SEEKING THE RELEVANT SOURCE

In this chapter, we illustrate by example how the algorithm works. In case that we need to find relevant documents that describe the concept we want to explicate, we first run our algorithm over all the documents that are at our disposal. As explained above, these documents are afterwards examined for their relevance.

In our example we will deal with the concept of *wild cat* so that we use our algorithm to obtain explications of the concept from the selected resources dealing with this concept. There are different explications of the same concept. For example, one resource describes wild cat from an anatomical perspective, another resource may describe the environment in which wild cats live, and still another document describes its behaviour.

The advantage of using the expressive apparatus of TIL is obvious here, since the analyses of sentences that mention the explicated concept are so fine-grained that they are easy to read and understand. Thus, users can easily analyse the differences between particular molecular concepts explicating the target concept and remove those documents that are semantically irrelevant for them. For instance, as mentioned above, if there are some inconsistencies between the so-obtained explications that the algorithm discovers, the user may exclude those that are not acceptable for him/her. Thanks to this approach, the selection is not based only on syntactic features like the occurrence of a given term, but also on semantic features provided by the fine-grained analysis.

Hypotheses are built up by applying the relation in intension *Typ-p* of type $(o(ol)_{\tau\omega}(ol)_{\tau\omega})_{\tau\omega}$ and the relation *Req* of type $(o(ol)_{\tau\omega})(ol)_{\tau\omega}$. *Typ-p* is the relation in intension between properties *P* and *Q* such that *typically*, if an individual happens to be a *Q* then most probably it has the property *P*. For example, the property of *having a weight between 1.2 and 11 Kg* is a typical property of the property of *being a wild cat*. On the other hand, *Req(uisite)* is a *necessary* relation between properties. Necessarily, if an individual happens to be a wild cat, then it must be a mammal as well.

In our example we had at our disposal three kinds of textual resources, namely one dealing with anatomic features, one with environmental features and finally reproduction features.

The anatomy resource:

- The weight of the wild cat is between 1.2 and 11 kilograms.
- The wild cat is a mammal.
- The wild cat has fur.
- The body length of the wild cat is 47 – 80 cm.
- The average skull capacity is 41.25 cm³.
- The average height of the wild cat at the withers is 37.6 cm.

The analysis comes down to these constructions.

$$\begin{aligned}
 \lambda w \lambda t \left[\left['Typ-p_{wt} [\lambda w \lambda t \lambda x [\right. \right. \\
 \quad \left. \left. [\leq ['Weight_{wt} x] '11] \wedge [\geq ['Weight_{wt} x] '1.2]] \right. \right. \\
 \quad \left. \left. ['Wild 'Cat] \right] \right. \\
 \wedge ['Req 'Mammal ['Wild 'Cat]] \\
 \wedge ['Req 'Has-Fur ['Wild 'Cat]] \\
 \wedge ['Typ-p_{wt} [\lambda w \lambda t \lambda x [\\
 \quad \left. \left. [\leq [['Average 'Body-Length]_{wt} x] '80] \wedge \right. \right. \\
 \quad \left. \left. [\geq [['Average 'Body-Length]_{wt} x] '47]] \right. \right. \\
 \quad \left. \left. ['Wild 'Cat] \right] \right. \\
 \wedge ['Typ-p_{wt} [\lambda w \lambda t \lambda x \\
 \quad \left. \left. [= [['Average 'Skull-Size]_{wt} x] '41.25] \right. \right. \\
 \quad \left. \left. ['Wild 'Cat] \right] \right. \\
 \wedge ['Typ-p_{wt} [\lambda w \lambda t \lambda x \\
 \quad \left. \left. [= [['Average 'Height]_{wt} x] '37.6] ['Wild 'Cat] \right] \right] \right]
 \end{aligned}$$

A resource dealing with the environment in which the wild cats live:

- Typical occurrence of wild cats is mixed or deciduous forests.
- The size of the territory of the wild cat is greater than 50 ha.
- The wild cat marks its territory with its claws, urination, droppings.

$$\begin{aligned}
 \lambda w \lambda t \left[\left['Typ-p_{wt} [\lambda w \lambda t \lambda x \right. \right. \\
 \quad \left. \left. ['Live-in_{wt} x [\lambda w \lambda t \lambda x [[['Mixed 'Forests]_{wt} y] \vee \right. \right. \\
 \quad \left. \left. [['Deciduous 'Forests]_{wt} y]]] \right] \right] ['Wild 'Cat] \right] \\
 \wedge ['Typ-p_{wt} [\lambda w \lambda t \lambda x \\
 \quad \left. \left. [\geq ['Territory-Size_{wt} x] '50] ['Wild 'Cat] \right] \right] \\
 \wedge ['Typ-p_{wt} [\lambda w \lambda t \lambda x [\\
 \quad \left. \left. ['Ter-Marking_{wt} x 'Clawing] \vee \right. \right. \\
 \quad \left. \left. ['Ter-Marking_{wt} x 'Urinating] \vee \right. \right. \\
 \quad \left. \left. ['Ter-Marking_{wt} x 'Leaves-Droppings] \right] \right] \\
 \left. \left. ['Wild 'Cat] \right] \right]
 \end{aligned}$$

Reproduction source:

- The in-heat period of the wild cat is 2 – 8 days.
- The wild cat is looking for a mate with a loud meow.
- The pregnancy period of a wild cat is 65 days.
- The size of the litter of wild cats is 3 – 4 kittens.

$$\lambda w \lambda t \left[\left[\begin{array}{l} 'Typ-p_{wt} [\lambda w \lambda t \lambda x [\\ \quad [\leq [In-Heat-Period_{wt} x] '8] \wedge \\ \quad [\geq [In-Heat-Period_{wt} x] '2]]] ['Wild 'Cat] \end{array} \right] \right. \\ \wedge \left[\begin{array}{l} 'Typ-p_{wt} [\lambda w \lambda t \lambda x \\ \quad ['Seek_{wt} x 'Mate ['Loud 'Meow]]] ['Wild 'Cat] \end{array} \right] \\ \wedge \left[\begin{array}{l} 'Typ-p_{wt} [\lambda w \lambda t \lambda x \\ \quad ['= [Pregnancy-period_{wt} x] '65]] ['Wild 'Cat] \end{array} \right] \\ \wedge \left[\begin{array}{l} 'Typ-p_{wt} [\lambda w \lambda t \lambda x [\\ \quad [\leq [Litter-Size_{wt} x] '4] \wedge \\ \quad [\geq [Litter-Size_{wt} x] '3]]] ['Wild 'Cat] \end{array} \right] \left. \right]$$

Types:

Weight, Body-Length, Height, Skull-Size, Territory-Size, Litter-Size, In-Heat-Period, Pregnancy-Period/ $(\tau \iota)_{\tau \omega}$: attributes
Average/ $((\tau \iota)_{\tau \omega}(\tau \iota)_{\tau \omega})$: attribute modifier
Mammal, Cat, Has-Fur; Forests, Clawing, Urinating, Leaves-Droppings, Mate, Meow/ $(o \iota)_{\tau \omega}$: properties
Wild, Kitten, Loud, Mixed, Deciduous/ $((o \iota)_{\tau \omega}(o \iota)_{\tau \omega})$: property modifiers
 $x \rightarrow \iota$
Live-in, Ter-Marking/ $(o \iota)(o \iota)_{\tau \omega} \tau \omega$
Seek/ $(o \iota)(o \iota)_{\tau \omega}(o \iota)_{\tau \omega} \tau \omega$

VII. CONCLUSION

In this paper, we described an application of the supervised machine learning methods to natural language processing. The goal was to explicate a given target concept by analysing text documents that mention the concept. We introduced an algorithm for the support in finding a relevant text resource. By applying methods of supervised machine learning adjusted to natural language processing by Transparent Intensional Logic, the sought concept is explicated by means of all the text resources submitted by the user who can afterwards decide which of the explications are best for his/her purpose.

As a future research, we plan to improve the post-processing of the obtained explications. In particular, we will concentrate on logical examination of the results; as outlined above, we want to detect inconsistencies and incompatibilities, to indicate potential fake news and intentional disinformation. Since we work hand-in-hand with linguists, the filtering can also be based on linguistic information like whether the target concept is denoted by a verb, noun, definite description or adjective. In case of verbs, we will also take into account verb valencies that indicate types of the arguments of the property denoted by a given verb.

ACKNOWLEDGMENT

This research has been supported by the Grant Agency of the Czech Republic, project No. GA18-23891S, "Hyperintensional Reasoning over Natural Language Texts" and also by the internal grant agency of VSB-Technical University Ostrava, project No. SP2019/40, "Application of Formal Methods in Knowledge Modelling and Software Engineering II". This work was also supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II) project IT4Innovations Excellence in Science – LQ1602, and by the Silesian University in Opava under the Student Funding Scheme, project SGS/11/2019.

REFERENCES

- [1] M. Duží, B., Jespersen, P. Materna, "Procedural Semantics for Hyperintensional Logic. Foundations and Applications of Transparent Intensional Logic.", Berlin: Springer, 2010.
- [2] M. Medveď, T., Šulganová, A., Horák, "Multilinguality Adaptations of Natural Language Logical Analyzer". In Proceedings of the Eleventh Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2017. Brno: Tribun EU, pp. 51-58, 2017.
- [3] M. Menšík, M. Duží, A. Albert, V. Patschka, M. Pajr, M., "Refining concept by machine learning." Computación y Sistemas, in press.
- [4] M. Menšík, M. Duží, A. Albert, V. Patschka, M. Pajr, M., "Machine learning using TIL". In Frontiers in Artificial Intelligence and Applications, Amsterdam: IOS Press, in press
- [5] T. M. Mitchell, "Machine learning." New York: McGraw-Hill, 1997. ISBN 00-704-2807-7.
- [6] Y. N. Moschovakis, "Sense and denotation as algorithm and value." In Lecture Notes in Logic, eds. J. Väänänen and J. Oikkonen, vol. 2, pp. 210-249. Berlin: Springer, 1994.
- [7] D. L. Poole, A. K. Mackworth, "Artificial intelligence: foundations of computational agents. 2nd pub.", Cambridge: Cambridge University Press, 2010. ISBN 978-0-521-51900-7.
- [8] S. J. Russell, P. Norvig, "Artificial intelligence: a modern approach.", 2nd ed. Harlow: Pearson Education, 2014. ISBN 978-1-29202-420-2.
- [9] P. Tichý, "Intensions in terms of Turing machines.", *Studia Logica* 26, pp. 7-25, 1969. Reprinted in (Tichý 2004: 93-109).
- [10] P. Tichý, "The Foundations of Frege's Logic.", Berlin, New York: De Gruyter, 1988.
- [11] P. Tichý, "Collected Papers in Logic and Philosophy", eds. V. Svoboda, B. Jespersen, C. Cheyne. Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press, 2004.
- [12] P. H. Winston, "Artificial Intelligence." 3rd ed., Mass.: Addison-Wesley Pub. Co., 1992. ISBN 02-015-3377-4.

A comparison of refinement techniques for the backtracking algorithms for the subgraph isomorphism problem

Jurij Mihelič

Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
jurij.mihelic@fri.uni-lj.si

Uroš Čibej

Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
uros.cibej@fri.uni-lj.si

Abstract—A well-known \mathcal{NP} -hard problem from graph theory is the subgraph isomorphism problem where the goal is for given pattern and target graphs to determine whether the pattern is a subgraph of the target graph. A plethora of algorithms for solving the problem exist in literature and most of them are based on the backtracking approach. In this paper, we explore several such algorithms. In particular, we focus on algorithmic techniques as well as code tuning and optimization approaches with the goal of refining basic backtracking algorithms. We also experimentally compare the performance of algorithms in order to evaluate and demonstrate their usefulness in practice.

Index Terms—subgraph isomorphism, algorithm, optimization, code tuning, experiment, evaluation

I. INTRODUCTION

Data structured in the form of graphs appear in many disciplines such as chemistry, biology, social networks, and document analysis. As the amount of data is increasing, its analysis requires better and better algorithms. In this paper we focus on a well-known \mathcal{NP} -hard combinatorial problem from graph theory, called *subgraph isomorphism problem*, where the goal is for given pattern and target graphs to determine whether the pattern is a subgraph of the target graph.

The problem has numerous applications and a plethora of algorithms for solving the problem exactly exist in literature. Most of them are based on the *backtracking* approach which explores the search tree of all possible solutions. As the problem is \mathcal{NP} -hard, all such algorithms exhibit exponential running time in the worst case. Nevertheless, in many practical cases, the problem can be quite efficiently solved if advanced techniques of pruning the search tree are employed.

In this paper, we focus on such techniques (intended for backtracking algorithms) which may provide a significant performance boost to algorithms especially if used in a combination with others. The techniques are mostly based on heuristic approaches as well as on the optimization and tuning of algorithms. They often originate from the field of constraint satisfaction and represent a basic approach in the field of algorithm engineering [10], [12]. The most widely used and well-known subgraph isomorphism algorithms are Ullmann's

algorithm [5], [14], VF variants [2], [3], [7], RI [1], Glasgow subgraph solver [9], FocusSearch [15], and LAD [13].

In the rest of the paper, we first formally define the subgraph isomorphism problem and present the backtracking approach to solving it. Afterward, we present backtracking refinement techniques. For each technique, we give a description followed by its experimental evaluation. Finally, we conclude the paper.

II. PRELIMINARIES

A. Graph Notions

We start this section with the definitions of several graph theory notions. A pair $G = (V_G, E_G)$, where $V_G = \{1, 2, \dots, n_G\}$ is a finite set of *vertices* and $E_G \subseteq V_G \times V_G$ is a set of vertex pairs representing *edges*, is called a *graph*. If edges are unordered or ordered then the graph is undirected or directed, respectively. In this paper, we mostly deal with undirected graphs, but our discussion is generally applicable also to the directed ones.

For an undirected graph $G = (V_G, E_G)$ and a vertex $u \in V_G$ we define

$$\mathcal{N}_G(u) = \{v \in V_G \mid \langle u, v \rangle \in E_G\}$$

and

$$d_G(u) = |\mathcal{N}_G(u)|$$

as the *u's neighborhood* and *degree*, respectively.

Let $H = (V_H, E_H)$ be a graph. A graph $G = (V_G, E_G)$ is a *subgraph* of a graph H if

$$V_G \subseteq V_H \quad \text{and} \quad E_G \subseteq E_H.$$

If it also holds

$$\forall u, v \in V_G : \langle u, v \rangle \in E_G \iff \langle u, v \rangle \in E_H$$

then the graph G is an *induced subgraph* of the graph H .

B. Morphisms

Let G and H be two graphs. Oftentimes, the graph G is called a *pattern* and the graph H a *target*. A bijective mapping $\phi : V_G \rightarrow V_H$ is called a *graph isomorphism* if

$$\forall u, v \in V_G : \langle u, v \rangle \in E_G \iff \langle \phi(u), \phi(v) \rangle \in E_H.$$

An injective mapping $\phi : V_G \rightarrow V_H$ is called *graph monomorphism* if

$$\forall u, v \in V_G : \langle u, v \rangle \in E_G \implies \langle \phi(u), \phi(v) \rangle \in E_H.$$

Instead of the term graph monomorphism the term (*ordinary*) *subgraph isomorphism* is often used since the mapping ϕ is an isomorphism between G and a subgraph $\phi(G)$ of the graph H . Here, $\phi(G)$ represents a subgraph of H to which G is mapped.

Thus, graph monomorphism is an injective mapping preserving adjacency relation where the images of vertices of graph G can have some additional edges in H not present in G . A more restricted version of the problem is called *induced subgraph isomorphism* which asks for an injective mapping $\phi : V_G \rightarrow V_H$ where

$$\forall u, v \in V_G : \langle u, v \rangle \in E_G \iff \langle \phi(u), \phi(v) \rangle \in E_H.$$

Consider a simple example in Figure 1 demonstrating the difference between the ordinary and induced version of the subgraph isomorphism.

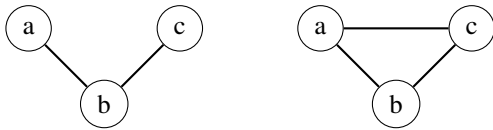


Fig. 1. The left graph G has no induced subgraph isomorphisms in the right graph H while it has six ordinary subgraph isomorphisms. Here, in the latter, the vertices abc of G may map to $abc, cba, bca, acb, bac,$ and cab of H .

C. Problem Definitions

Several versions of the subgraph isomorphism problem exist:

- (decision problem) Given graphs G and H , is there a subgraph isomorphism between the two?
- (enumeration problem) Given graphs G and H , list all the subgraph isomorphisms between the two.
- (counting problem) Given graphs G and H , count the number of subgraph isomorphism between the two.

All these problems can be about ordinary or induced subgraph isomorphisms.

To show the \mathcal{NP} -completeness of the decision version of the problem, a reduction from the well-known clique or Hamiltonian cycle problems is usually employed. See also [6] for an overview of the problem.

D. Backtracking Algorithms

Many of the algorithms for the subgraph isomorphism problem are based on the backtracking approach. Here, the mapping $\phi : V_G \rightarrow V_H$ is constructed gradually where on each step one assignment of a vertex $u \in V_G$ is resolved, i.e., $\phi(u)$ is assigned to some $u' \in V_H$. Without loss of generality, we assume (if not otherwise noted) that the vertices of V_G are processed in the order of their labels, i.e., $1, 2, \dots, n_G$.

A state where some vertices of V_G are already assigned is called a *partial solution*. In particular, on step k , where $1 \leq k \leq n_G$ the algorithm resolves the vertex $k \in V_G$ and constructs a partial solution ϕ_k from ϕ_{k-1} . Here, $\phi_0(u) = \perp$ and $\phi_k(u) = \phi_{k-1}(u)$ for all $u < k$. Finally, $\phi(u) = \phi_{n_G}(u)$.

A *consistency* of a partial solution ϕ_k may be checked by satisfying the following constraints:

- (injectivity) $\phi(u) \neq \phi(v)$ for each $u, v \in V_G : u \neq v$,
- (adjacency) $\langle \phi(u), \phi(v) \rangle \in E_H$ for all $\langle u, v \rangle \in E_G$, and
- (non-adjacency) $\langle \phi(u), \phi(v) \rangle \notin E_H$ for all $\langle u, v \rangle \notin E_G$.

The last constraint is only relevant to the induced version of the problem. Notice that, other constraints such as matching of labels of vertices or edges may also be present.

There are two general approaches to how to implement candidate checking in backtracking algorithms:

- (backward-checking) When a new candidate vertex $u' \in V_H$ is mapped to some vertex $u \in V_G$ the consistency of the partial solution is checked: if the check fails, then the algorithm backtracks, otherwise, it proceeds to the next vertex of G . When the last vertex is successfully mapped the algorithm finds a subgraph isomorphism.
- (forward-checking) Whenever a vertex $u \in V_G$ is mapped to $u' \in V_H$ we use this to form additional constraints on candidates for vertices not yet mapped. For example, due to injectivity, no vertex in V_G can be mapped to u' . Such additional constraints may be stored in a binary matrix where the u -th row gives the u 's candidates.

All of the algorithms listed in the introduction emphasize on one of these two approaches.

III. REFINEMENT TECHNIQUES

In this section, we present several refinement techniques applicable to backtracking algorithms for the subgraph isomorphism problem.

A. Search Order

One of the most widely used refinements of backtracking algorithms is the exploitation of the order in which the vertices are processed. In this section, we briefly explore this option while we refer the reader to [4] for a more in-depth analysis.

The backtracking algorithm correctness does not depend on the ordering, but the shape of a search tree does; smaller trees may result in much more efficient algorithms. In particular, the goal is to use such an order that leads to inconsistency as soon as possible in order to prune larger parts of the tree.

Here, we present several heuristics for ordering the vertices. Let $(u_1, u_2, \dots, u_{k-1})$, where $u_i \in V_G$, be a partial

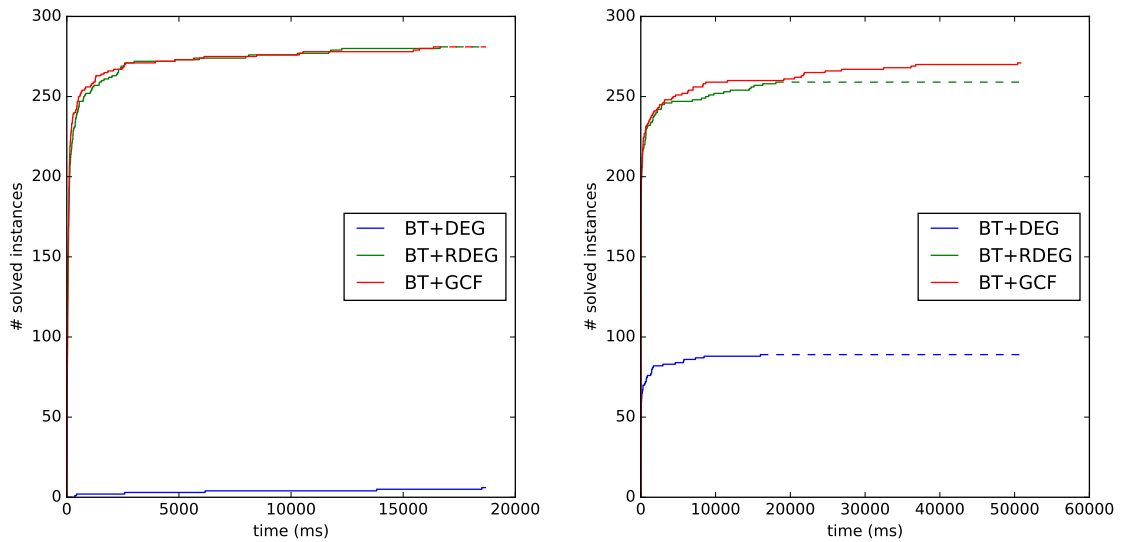


Fig. 2. Results of the experiments with the backward-checking backtracking algorithm and various vertex orders on random graphs (left) and bounded valence graphs (right): x -axis specifies time-allowed for solving and y -axis gives the number of solved instances in the given time.

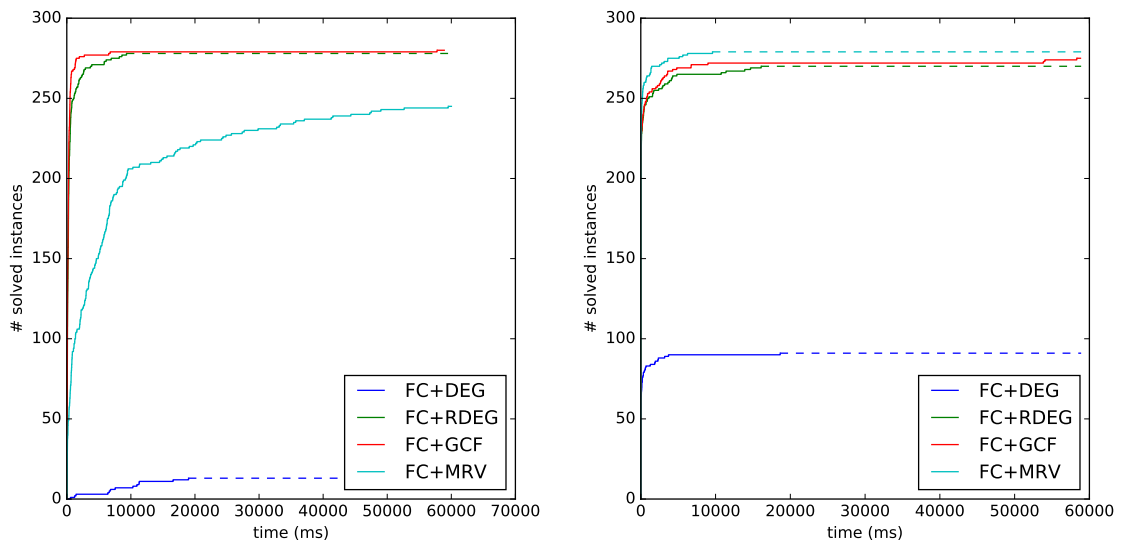


Fig. 3. Results of the experiments with the forward-checking backtracking algorithm and various vertex orders on random graphs (left) and bounded valence graphs (right): x -axis specifies time-allowed for solving and y -axis gives the number of solved instances in the given time.

ordering of vertices. In what follows we also denote $U_{k-1} = \{u_1, u_2, \dots, u_{k-1}\}$ and present several options for selecting the next vertex $u_k \in V_G \setminus U_{k-1}$:

DEG (degree) Vertices are non-increasingly ordered by degree, i.e.,

$$u_k = \arg \max_{u \in V_G \setminus U_{k-1}} d_G(u).$$

RDEG (relative degree) Start with a vertex with the highest degree. Afterward we proceed with vertices which are adjacent to the maximal number of already se-

lected vertices, i.e.,

$$u_k = \arg \max_{u \in V_G \setminus U_{k-1}} |\mathcal{N}_G(u) \cap U_{k-1}|.$$

If several vertices match, we select one with a higher degree.

GCF (greatest constraint first) This order (see also [1]) is similar to RDEG but equalities are resolved differently. Let M be the set of vertices with at least one neighbor in the partial order, i.e.,

$$M = \{u \in V_G \setminus U_{k-1} \mid \mathcal{N}_G(u) \cap U_{k-1} \neq \emptyset\}$$

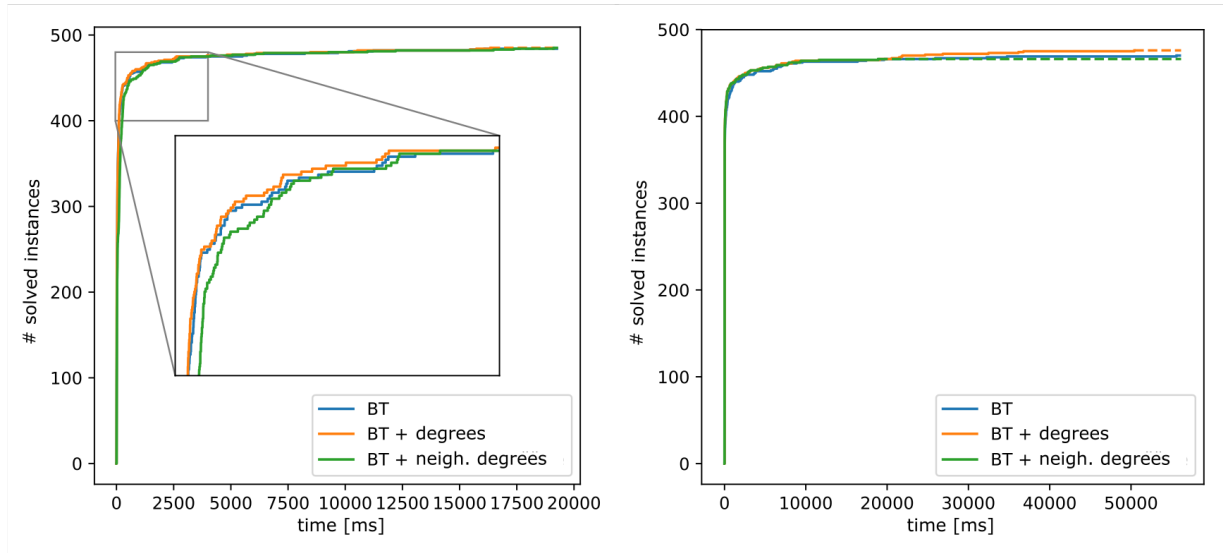


Fig. 4. Results of the experiments for the algorithms using degree constraints on random graphs (left) and bounded valence graphs (right): Solved test instances – degree constraints: x -axis specifies time-allowed for solving and y -axis gives the number of solved instances in the given time.

and the next selected vertex

$$u_k = \arg \max_{u \in V_G \setminus U_{k-1}} (|\mathcal{N}_G(u) \cap U_{k-1}|, |\mathcal{N}_G(u) \cap M|, d_G(u)).$$

MRV (minimal remaining values) In the forward checking algorithms we can, for each vertex $u \in V_G$ store its candidate vertices from V_H . Using this information we can, in each state of the search tree, select such a vertex that has a minimum number of candidates. The goal is to minimize the branching of the search tree.

To compare the performance of the above orders we performed an experiment. See Fig. 2 for the comparison of orders for backward-checking algorithms and Fig. 3 for the comparison of orders for forward-checking algorithms. Observe that, DEG order (which is the most straightforward) performs inferior to the other orders. Using more advanced orders significantly improves the performance of both backward and forward-checking algorithms.

B. Degree Constraints

Another approach to pruning the search tree is to derive additional problem constraints. To do this we use basic (sufficient) constraints given by the problem definition. The goal is that the derived constraints can be efficiently checked and if they are not satisfied we can prune the tree.

First, let us derive the *vertex degree* constraints. Since the mapping of vertices is an injection, we can see that

$$d_G(u) \leq d_H(\phi(u)) \quad \text{for each } u \in V_G.$$

Indeed, the neighborhood of $u \in V_G$ must be mapped to the neighborhood of $\phi(u)$. Thus, $\phi(\mathcal{N}_G(u)) \subseteq \mathcal{N}_H(\phi(u))$ and $|\mathcal{N}_G(u)| \leq |\mathcal{N}_H(\phi(u))|$.

Explicitly storing vertex degrees in the graph data structure enables us to very efficiently check degree constraints. Notice that, in the backward checking the constraint is easily checked for every new mapping made while in the forward checking the constraint is included in the preprocessing phase.

Degree checking can be extended also to a neighborhood of a vertex. First, we define a *neighborhood degree sequence* $Z_G(u)$ for a vertex $u \in V_G$ as

$$Z_G(u) = (d_G(v))_{v \in \mathcal{N}_G(u)} \quad \text{sorted nonincreasingly.}$$

Observe that, for each vertex $u \in V_G$ it must hold

$$Z_G(u) \preceq Z_H(\phi(u)) \quad (1)$$

where \preceq is the relation of lexicographical comparison of two sequences.

The constraint is a consequence of the vertex degree constraint. In particular, we can see that for each $v \in \mathcal{N}_G(u)$ there must be a unique vertex $v' \in \mathcal{N}_H(\phi(u))$ such that $d_G(v) \leq d_H(v')$. Furthermore, checking this for the whole neighborhood corresponds to Eq. (1).

To compare the performance of both refinements we performed an experiment with the backtracking algorithm. See Fig. 4 for the results. Observe that, using degree constraints slightly improves the performance of the algorithms while neighborhood degree constraints seem to be too much of an overhead to improve the total performance.

C. Bit arrays

In the subgraph-isomorphism backtracking algorithms, oftentimes, set operations such as union and intersection are used, where sets are subsets of vertices of graph G (or H). One way to efficiently represents such sets is to use *bit arrays*, where the i -th bit of the array corresponds to the membership of the vertex i (we assume vertices are numbered from 1 to

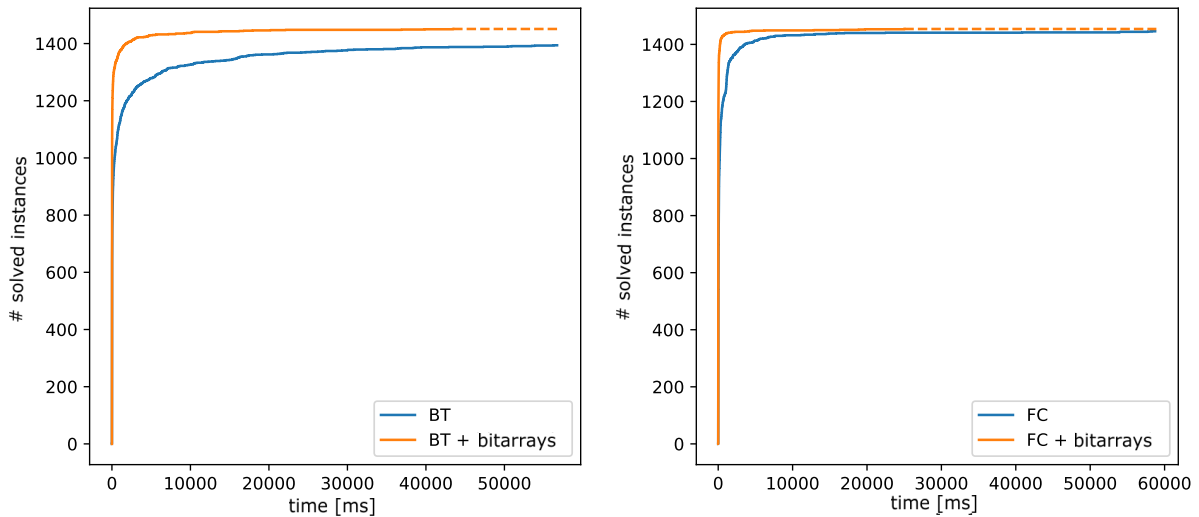


Fig. 5. Results of the experiments on random graphs for the algorithms using bitarrays and backward-checking (left) and forward-checking (right): x -axis specifies time-allowed for solving and y -axis gives the number of solved instances in the given time.

n). Now using a bit array representation, union correspond to bitwise or, intersection to bitwise and and difference to and and not.

A graph can be represented with bit arrays, where for each vertex a bit array is stored representing a set of the vertex's neighbors. To compute the set of candidates for the vertex $u \in V_G$ we consider only the vertices from the intersection of neighborhoods of already mapped neighbors of u , i.e.,

$$\bigcap_{v \in \mathcal{N}_G(u), v < u} \mathcal{N}_H(\phi(v)).$$

To see the practical effect of using bit arrays see Fig. 5. We observe that in both forward and backward checking algorithms the use of bit arrays has a positive effect on the running time.

D. Parents-based Candidate Selection

In this section, we describe an algorithm tuning technique which does not prune the search tree but it enables a faster selection of candidate vertices. Let $(u_1, u_2, \dots, u_{k-1})$, where $u_i \in V_G$, be a partial ordering of vertices and let $u_k \in V_G$ be the next vertex to be assigned a vertex $u'_k \in V_H$. Moreover, let u_k be adjacent to some vertex u_p appearing before u_k in the search order, i.e., $1 \leq p < k$. We call such a vertex u_p , a *parent* of u_k .

Since a vertex $u'_k \in V_H$ to be assigned to $u_k \in V_G$ must also be adjacent to a vertex $u'_p \in V_H$ that is assigned to $u_p \in V_G$, we may, when selecting u_k 's candidate vertices, iterate only on the neighbors of u'_p , i.e., $u'_k \in \mathcal{N}_H(u'_p)$.

Notice that, it is possible that u_k does not have a parent, so the algorithm must still check all the vertices. However, in connected graphs the parent almost always exists; in particular, this is true for the RDEG ordering which orders the vertices according to the number of parents. Results of the

experimental evaluation are in Fig. 7 and are described in the next subsection.

E. Neighborhood Adjacency Checking

Let $u \in V_G$ and $u' \in V_H$ be the two vertices so that u is mapped to u' . We present another approach to consistency-constraint checking which actually probes the vertices in the neighborhoods of u and v . In particular, we check all the vertices $v \in \mathcal{N}_G(u)$ that are already mapped; we denote with $v' \in V_H$ a vertex in the target graph to which v is mapped. Notice that, a vertex v' (corresponding to v) must also be adjacent to u' (corresponding to u). See Figure 6 for a graphical representation of neighborhood adjacency checking. If this neighborhood adjacency check fails then the current mapping cannot represent a valid subgraph isomorphism.

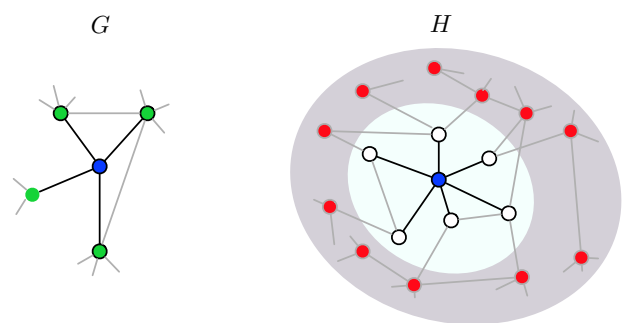


Fig. 6. Filtering based on the selected pattern and target vertices (blue). The neighbors (green) of the pattern vertex are incompatible with the non-neighbors (red) of the target vertex (color online).

Notice that, in the case of the induced subgraph isomorphism problem, the same constraint can also be checked from the perspective of the target vertex to pattern vertex

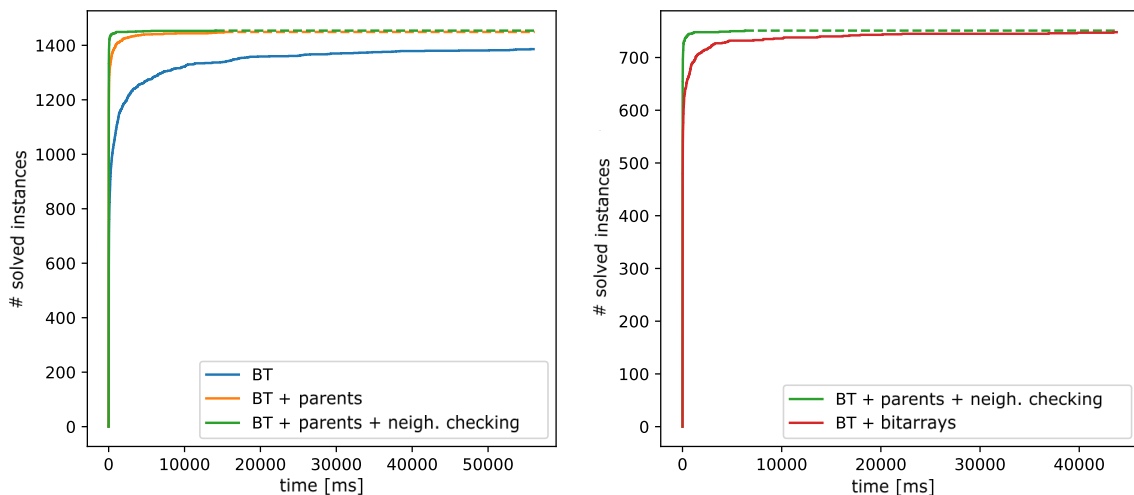


Fig. 7. Results of the experiments on random graphs for the algorithms using various refinements and their combinations: x -axis specifies time-allowed for solving and y -axis gives the number of solved instances in the given time.

mapping. To efficiently implement this technique we need to represent graphs with adjacency lists (for fast iteration of neighborhoods) as well as adjacency matrix (for fast checking of adjacency of two vertices). Moreover, we also have to store the inverse of the mapping function to efficiently obtain a vertex in the pattern which was mapped to a particular target vertex. See also [5] for details.

In Fig. 7 we show the results of our experimental evaluation. Here, we compare the improvements of using parents as well as neighborhood adjacency checking and bit arrays. One can observe that the combination of using parents together with neighborhood checking performs the best. We can also observe that bit arrays (which are unfortunately incompatible with neighborhood checking) are slower than the parent-neighborhood checking combination.

IV. CONCLUSION

In this paper, we presented and experimentally evaluated several refinements for the subgraph isomorphism problem. Of course, several other refinements are possible and are also a part of our future research agenda. To name just a few from the constraint satisfaction area there is the so-called *arc consistency* and *Hall sets* that seem promising. Furthermore, our previous research [8], [11] on graph symmetry exploitation (pattern and target automorphisms) on the so-called *exploratory equivalence* also may provide speedups in graphs exhibiting symmetries.

ACKNOWLEDGMENT

Our gratitude goes to Sven Cerk who performed most of the experimental work.

REFERENCES

- [1] Vincenzo Bonnici, Rosalba Giugno, Alfredo Pulvirenti, Dennis Shasha, and Alfredo Ferro. A subgraph isomorphism algorithm and its application to biochemical data. *BMC bioinformatics*, 14(7), 2013.
- [2] Vincenzo Carletti, Pasquale Foggia, Antonio Greco, Mario Vento, and Vincenzo Vigilante. Vf3-light: A lightweight subgraph isomorphism algorithm and its experimental evaluation. *Pattern Recognition Letters*, 125:591 – 596, 2019.
- [3] Vincenzo Carletti, Pasquale Foggia, Alessia Saggese, and Mario Vento. Introducing vf3: A new algorithm for subgraph isomorphism. In Pasquale Foggia, Cheng-Lin Liu, and Mario Vento, editors, *Graph-Based Representations in Pattern Recognition*, pages 128–139, Cham, 2017. Springer International Publishing.
- [4] Uroš Čibej and Jurij Mihelič. Search strategies for subgraph isomorphism algorithms. In Prosenjit Gupta and Christos Zaroliagis, editors, *Applied Algorithms*, pages 77–88, Cham, 2014. Springer International Publishing.
- [5] Uroš Čibej and Jurij Mihelič. Improvements to Ullmann’s algorithm for the subgraph isomorphism problem. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(07):1550025, 2015.
- [6] D Conte, P Foggia, C Sansone, and M Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004.
- [7] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372, Oct 2004.
- [8] Luka Fürst, Uroš Čibej, and Jurij Mihelič. Maximum exploratory equivalence in trees. In *2015 Federated Conference on Computer Science and Information Systems, FedCSIS 2015, Łódź, Poland, September 13-16, 2015*, pages 507–518, 2015.
- [9] Ciaran McCreesh and Patrick Prosser. A parallel, backjumping subgraph isomorphism algorithm using supplemental graphs. In Gilles Pesant, editor, *Principles and Practice of Constraint Programming*, pages 295–312, Cham, 2015. Springer International Publishing.
- [10] Catherine C. McGeoch. *A Guide to Experimental Algorithmics*. Cambridge University Press, 2012.
- [11] Jurij Mihelič, Luka Fürst, and Uroš Čibej. Exploratory equivalence in graphs: Definition and algorithms. In *2014 Federated Conference on Computer Science and Information Systems, Fedcsis 2014, Warsaw, Poland, September 7-10, 2014*, pages 447–456, 2014.
- [12] Matthias Muller-Hannemann and Stefan Schirra, editors. *Algorithm Engineering: Bridging the Gap Between Algorithm Theory and Practice*. Springer-Verlag, Berlin, Heidelberg, 2010.
- [13] Christine Solnon. AllDifferent-based filtering for subgraph isomorphism. *Artificial Intelligence*, 174(12-13):850–864, August 2010.
- [14] J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, January 1976.
- [15] Julian R. Ullmann. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *Journal of Experimental Algorithmics*, 15:1.6:1.1–1.6:1.64, February 2011.

Classification of Tire Tread Images by Using Neural Networks

1st Alžbeta Michalíková
 Faculty of Natural Sciences
 Matej Bel University
 Banská Bystrica, Slovakia
 alzbeta.michalikova@umb.sk

2nd Branislav Pažický
 Faculty of Natural Sciences
 Matej Bel University
 Banská Bystrica, Slovakia
 brano.pazicky@gmail.com

Abstract—Image classification represents a complex problem which occurs in variety of science and real life applications. Neural network is one of the excellent tools which could be used to solve this problem. The research presented in this paper is motivated by several consultations with active crime scene investigators. During these consultations, it became apparent, that criminology department is in critical need of advanced software for tire tread print identification. The current software used by crime scene investigators is outdated, time demanding and hard to work with. Application for tire tread print identification should be one of results of project created on the basis of our work with criminology department. This paper presents one of the first steps of this identification - tire tread position identification.

Index Terms—neural network, image classification, image pre-processing

I. INTRODUCTION

At the time recognition of images by using computers is one of the most important processes. There is dedicated research area, called computer vision, which specializes in finding solutions to this problem. The aim of this discipline is to imitate human vision with the use of computer as well as possible. If the person tries to recognize some object in the image, he/she is using a lot of criteria and properties such as intelligence, previous experiences and knowledge. If the computer is used for solving of this problem, all the criteria need to be programmed by a programmer.

In this article, we present the application which was developed for classification of the images of the tire treads into the predefined classes by using neural networks. The research presented in this paper is motivated by several consultations with active crime scene investigators. During these consultations, it became apparent, that criminology department is in critical need of advanced software for tire tread print identification. The current software used by crime scene investigators is outdated, time demanding and hard to work with. This led to development of new software tool for tire print identification. Prototype of this application is still being developed by department of Computer Science of Matej Bel University. The problem can be formalized as follows: While on the crime scene, criminologists often find various types of prints, such as foot prints, finger prints and one of most common types of prints - tire thread prints. If there is a possibility to recognize the brand or manufacturer of tires

present on the crime scene, they could be one of the important evidence which could contribute to the conviction of offender. To recognize the tire tread brand or manufacturer, the database of tire tread prints needs to be build [1], [2], [3]. The data for this database could be obtained by making the photos of tires for example at the concrete seller. By using this approach, one could make photos of high quality with the proper position of the tires, the proper lighting, etc. Of course this possibility is time consuming and resource demanding. On the other hand, the images of the tire treads could be obtained from the web pages. Not all images are in the required quality and in the proper position, but the access to these images is simpler. In our previous work, presented in [4], we described process of creating database of download tire thread images. We developed the web crawling application to extract the relevant images together with important information from the predefined web pages. Not all images are suitable for the additional processing and for comparing with the tire tread prints obtained at the crime scene. In [5] authors presented seven basic types/classes of the tire tread images and they described the advantages/disadvantages of each class from the point of comparison mentioned in the paper [5]. The next step of processing of the data from these obtained classes is the specification of the tire tread prints. The position of tire tread print is specific for each class of tire treads as presented on the Fig. 1. The process of obtaining these prints for one



Fig. 1. Visualization of position of tire tread print.

specific class by using the *Hough transform* and *fuzzy sets* was described in [6]. Authors of [7] used genetic algorithms to



Fig. 2. Six classes of tire tread images.

optimize the number of lines which are, in mentioned article, the main features for tire prints. After that the tire tread prints can be saved in the database and used for comparison with those, found at crime scene.

This paper presents one of the first steps of tire tread print identification - tire tread position identification. Precisely, we need to classify image into the classes on the basis of position of wheel on the image Fig. 2. In this article the images obtained from the web crawling application [4] were classified, but just six classes of tire treads were considered. Two classes with similar properties mentioned in [5] were merged together. Six classes of tire tread images used in this article are displayed on Fig. 2. The authors of the presented paper are interested in use of *soft computing tools* for solving of this problem. In the paper [5] authors developed an algorithm for classification based on *histogram of oriented gradients* and *support vector machine*. Of course, this classification was not one hundred percent accurate and therefore there is a space to develop another algorithms. For example, in [8] *intuitionistic fuzzy sets* and their *similarity measure* for classification images of tire treads were used. In this article, *neural networks* are chosen as the main technique for classification.

The pre-processing techniques which were used in this paper are described in Section II. In Section III the structure of *artificial neural network* (NN), the input and output data and the procedure of dividing images into the classes and also building of NN are presented. The notes to application which was developed in MATLAB are presented in Section IV. At the end of the paper, in Section V, the results of classification are presented and compared with classification mentioned in [8].

II. PRE-PROCESSING OF THE IMAGES

The images of tire treads obtained by our web crawling application from web pages are in different graphical formats, different sizes and colours. In the first step we need to transform them to such format, that they could be used by proposed algorithm. The process of data preparation is called pre-processing and in our application, which was developed in MATLAB, it consisted of five steps.

- *Converting images into the JPEG format.*
Since the images are downloaded from the different web pages they are in different graphical formats. Graphical format JPEG is one of the most used graphical formats, which was developed to displaying the photos and images. In this step each image from the obtained set of images is taken and converted to graphical format JPEG.
- *Converting images into the gray scale format.*
Some of the images are in truecolour format, some are in the gray scale format. Before the next step of algorithm all the images are converted into the gray scale format.
- *Removing white pixels on all sides of the image.*
The images of the tire treads are usually made on white background. We need to work just with the tire tread object therefore in this phase it is necessary to remove the white pixels on the sides of images, see Fig. 3. The algorithm is working in such way that:
 - the critical value is selected,
 - image is saved as a matrix,
 - in the cycle algorithm checks values in the rows (and then columns) of the matrix,
 - if all elements of the row (column) are greater then selected critical value, given row (column) is removed.



Fig. 3. Visualization of white pixel removal.

- *Converting images into the black and white format.*
The images are converted into the black and white format by using MATLAB function `imbinarize` where its

specialized method `adaptive` is applied. This function is used for calculating of the pixel value and statistical values of its neighbouring pixels. The differences between classical converting function and function `imbinarize` with `adaptive` method could be seen on the Fig. 4.

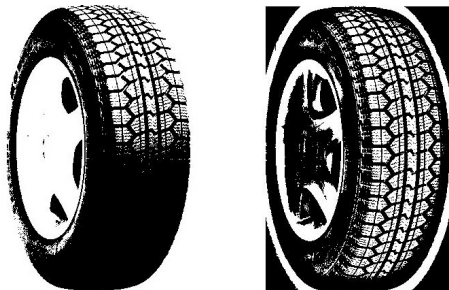


Fig. 4. The result of standard BW function (left) and function `imbinarize` (right).

- *Dividing image into 16 parts.*

As it was mentioned in the Introduction, the images of tire treads are divided into the six classes by using artificial neural networks (NN). As an input to NN, the normalized vectors with information about the number of white pixels in individual parts of image was chosen. For this system each image is divided into 16 parts as presented in the Fig. 5.



Fig. 5. Division of the image into the 16 parts.

The motivation for use of this dividing is such that if the image of whole wheel is taken, there are many white pixels in each corner of picture. On the other hand, on the images with half wheel objects, there are usually two corners with many white pixels and two of corners contains mostly black pixels. Similarly, one could see the differences in the central part of the images.

Note that the image in this part of the algorithm is in the same size as it was downloaded from the web page. Of course the size of image, i.e. the number of rows or columns, is not always possible to divide by number four. In such situation program will use created function which will gradually remove as many rows or columns (from 1 to 3) as needed. After that the matrix is divided into the 16 sub-matrices and for each new matrix the number of

white pixels is computed. The computed values of white pixels are saved into the 16 coordinates vectors. As the inputs to NN, the normalized vectors with values from interval $[0, 1]$ are expected. Therefore, last modification in this step is normalization of the coordinates of the vector. Note, that each sub-matrix of image matrix has the same size. Therefore the product of number of rows and number of columns of sub-matrix is computed and after that each coordinate of the vector is divided by this number.

Now image information which is stored as normalized vector is prepared for processing by neural network. Even though used images were not of the same size, the normalized vectors are in the same size. This way, we can extract as much information from image as possible. By converting images into the same size one could lost some information. Similarly, the filters for removing image noise were not used. The pre-processing of the images was done automatically and therefore it was not possible to choose the best noise filter for each image.

III. PROPOSED NEURAL NETWORK

Neural networks have the basis in biologic processes and they are used for simulation of the functions of human neural system. They can be also used for classification. For design and implementation of neural network we used Neural Network Toolbox, which is one of the libraries of MATLAB. This toolbox allows us to create different types of neural networks and allows quick and effective processing of sizable data sets.

When building neural network, it is important to prepare data. The higher the number of data samples is, the better results could be expected. Therefore, the set of 1293 images was prepared for building of the neural network. It is required to separately prepare input and output matrix. The input matrix is a matrix where the images are described, so it consists of the vectors with 16 coordinates and size of this matrix was 16×1293 . On the other side output matrix represents the groups into which the images need to be classified. Since we considered six classes, output matrix was of the size 6×1293 , where each column consists of five values equal to 0 and one value equal to 1. The value 1 was on the position which represented the pertaining class. Various architectures for training neural networks was chosen. The aim was to use as simple network as possible with satisfactory results. Considering the structure of data, the feedforward neural network with one hidden layer, which consisted of 15 neurons was used. On the Fig. 6 we present the structure of proposed neural network. To divide the data into the training, testing and validation set the default ratio was used (training : testing : validation = 0.7 : 0.15 : 0.15). After the building of the network we obtained following results:

- 93.4 percent of training data were classified correctly,
- 89.7 percent of testing data were classified correctly,
- 90.2 percent of validation data were classified correctly,
- 92.3 percent represents the correctness of data classification in the created neural network.

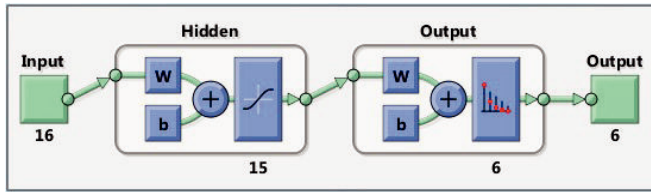


Fig. 6. The structure of proposed artificial neural network.

IV. DESCRIPTION OF THE DEVELOPED CLASSIFICATION APPLICATION

As mentioned before the aim of this study was to create automatic system which takes images from the obtained set of images, modifies them to the required numerical form and by using created classification system, which is based on neural networks, classifies the images into the groups. Each group represent position of tire tread on the image. For automatic classification of the images the algorithm in MATLAB was created. It consists of following steps:

- *Creation of new folders.*
For the easy verification of the success of classification the program was designed in such way, that after the classification the images were moved to the pertaining folder. Since there were considered six classes algorithm created six new folders.
- *Loading images and converting them into the vectors.*
This step was described in the Section II. The result of this step is 16 coordinate normalized vector for each image.
- *Classification of the selected images into the groups.*
The selected images are processed by created neural network system. After this step the information about the group assignment of each image is known.
- *Migration of the images into the created folders.*
The last step of the algorithm is migration of images into the created folders due to the groups which were obtained in previous step.

The user of the application need to give to system some folder with images. System automatically takes images one by one from this folder, classified them and migrate them to the new designed folders. After the end of the running of algorithm it is enough to open individual designed folders and control the results of classification.

V. RESULT OF CLASSIFICATION AFTER THE USE OF SYSTEM BASED ON NEURAL NETWORKS

As mentioned in the introduction, there were studies aimed at the classification of tire tread images done before. Since we didn't get the set of images from the first mentioned study [5] we could compare results with the results of second study [8]. In this study we used same 326 samples as in [8]. In the Table I there are results of classification system created by using neural network. From this table, one can compute that

TABLE I
RESULT OF CLASSIFICATION

Class	Correct	Incorrect	Sum
Class 1	122	6	128
Class 2	48	3	51
Class 3	37	11	48
Class 4	51	1	52
Class 5	8	0	8
Class 6	34	5	39
Sum	300	26	326

92.02 percent correctness of image classification was reached. In the study [8] which was done on the same data, the success of classification reached 90.8 percent. We can conclude that the improvement was reached.

If we look at the images which were classified incorrectly, we could conclude following observations.

- The incorrectly classified images are images with insufficient lighting. This caused that some images are classified into the wrong class, mostly symmetric with the correct class. For example the image on Fig. 7a was classified into the Class 2. The correct class for this image is Class 1 (see Fig. 2).
- The second reason of wrong classification is that the images of tire tread prints are situated in such position that even people have problem to classify them to one specific class. For example the image on Fig. 7b was classified into the Class 3 and at the same time the image on Fig. 7c was classified into the Class 1. As we can see Fig. 7b and Fig. 7c display part of the wheel, which can't be classified as whole wheel or half of the wheel (Class 1 or Class 3 on Fig. 2).
- Moreover, in the study presented in [8] we used specialized class which contains the images with text. We didn't consider this class therefore images which contain text, were classified into the one of the considered 6 classes. There were 7 of such images. These images contribute to decrease of correctness of classification.



Fig. 7. Examples of wrong classified images.

VI. FUTURE WORK

As mentioned in the introduction, after the classification of the images into the selected classes, the rectangles with tire

print images are selected on the basis of their class assignment. The rectangles with tire tread print images are added into the database and by the different techniques they are matched with the prints which were found at crime scene. This represents a complex problem and huge amount of work. During the study of literature, the authors found some interesting articles which paid attention to the same or similar problems. For example, in [7] the genetic algorithms were used to optimize the number of lines which are, in mentioned article, the main features for tire prints. In our future work, the genetic algorithms could be used for optimization of the parameters of neural network and in such way they could help to improve obtained results.

ACKNOWLEDGMENT

This work was supported by the grant KEGA 025UMB-4/2017.

The support of the Slovak Research and Development Agency under the contract No. APVV-0219-12 is kindly announced.

REFERENCES

- [1] W. J. Bodziak, "Forensic footwear evidence," CRC Press, 2017.
- [2] F. H. Lux, "Tire Track Identification", *Journal of Forensic Research*, ISSN: 2157-7145, vol. 4:198. doi:10.4172/2157-7145.1000198, 2013.
- [3] P. Poquet Domenech, "Tire print identification through registration techniques and attribute graphs," MS thesis. Universitat Politcnica de Catalunya, 2013.
- [4] M. Vagač, M. Melicherčík, M. Marko, P. Trhan, A. Michalíková, R. Kliment and R. Drapka, "Crawling images with web browser support," 13th International IEEE Scientific Conference on Informatics'2015, pp. 286–289, 2015.
- [5] M. Vagač, M. Melicherčík and J. Schon, "Classification of tire images in order to obtain the best possible tire tread sample," The 5th International Scientific Conference, Applied Natural Science 2015. September 30-October 2, 2015, Jasn; Trnava : UCM, p. 173, 2015.
- [6] A. Michalíková and M. Vagač, "A tire tread pattern detection based on fuzzy logic," *Flexible Query Answering Systems 2015*. Springer, Cham, pp. 381–388, 2016.
- [7] E. Acevedo, A. Acevedo, F. Felipe and P. Avils, "Artificial intelligence tools for pattern recognition." Second International Workshop on Pattern Recognition. Vol. 10443. International Society for Optics and Photonics, 2017.
- [8] A. Michalíková, "Intuitionistic fuzzy sets and their use in image classification," *Notes on Intuitionistic Fuzzy Sets*. Academic Publishing House, Sofia, Bulgaria. Volume 25, Number 2, 2019, p. 60-66. Print ISSN 1310-4926, Online ISSN 2367-8283.

High Order Numerical Integration Method and its Applications – The First 36 Years

Gabriela Nečasová*, Petr Veigend* and Václav Šátek*†

*Brno University of Technology, Faculty of Information Technology

Božetěchova 2, 612 66

Brno, Czech Republic

(inecasova,iveigend,satek)@fit.vutbr.cz

†IT4Innovations, VŠB Technical University of Ostrava

17. listopadu 15/2172, 708 33

Ostrava-Poruba, Czech Republic

vaclav.satek@vsb.cz

Abstract—The aim of this paper is to remember the work of our colleague Jiří Kunovský. The high order integration method that he developed with his students and colleagues has many positive properties and is studied to this day. The paper shows these properties on a set of technical initial value problems. The MATLAB implementation of Modern Taylor Series Method is compared with state-of-the-art MATLAB solvers.

Index Terms—ordinary differential equations, initial value problems, Taylor series, MTSM, MATLAB

I. INTRODUCTION

Since 1983, Jiří Kunovský as a lecturer in Department of Automatic Computers at Brno University of Technology started to focus on electrical circuits and logic circuits as an engineer. He found out the convenience of describing such systems by differential equations. Using programmable units, he created a unique algorithm to simulate those systems using simplified form.

He worked on several projects financed by national funds (e.g. Parallel implementation of numerical methods for large systems, Efficient application of stiff and high-order systems) and cooperated on projects with other foreign institutions (e.g. TU Wien, University of Auckland). He had published more than 120 papers during his career [1].

The paper deals with the solution of technical initial value problems (IVPs). The solution of the technical IVPs means to find the numerical solution of the system of ordinary differential equations (ODEs). In the paper, ODEs are solved using a high order variable-step variable-order numerical integration method – Modern Taylor Series Method (abbreviated as MTSM in the paper).

The paper consists of several sections. In section II, MTSM and its properties is introduced. Subsections II-A and II-B focus on the solution of linear and nonlinear problems using MTSM, respectively. Several examples of linear and nonlinear problems are introduced in section III, namely movement of the charged particle III-A, simple pendulum III-B and calculation of Fourier coefficients III-C. The results obtained by MTSM are compared with the state-of-the-art MATLAB solvers.

II. MODERN TAYLOR SERIES METHOD

This section of the paper briefly introduces the developed method – MTSM. It compiles information from previously published works, mainly [2], [3], [4] and other works cited throughout.

An ODE

$$y' = f(t, y), \quad (1)$$

with initial condition

$$y(t_0) = y_0, \quad (2)$$

is called an initial value problem. The best-known and the most accurate method of calculating a new value of the numerical solution of ODE is to construct the Taylor series in the form

$$y_{i+1} = y_i + h \cdot f(t_i, y_i) + \frac{h^2}{2!} \cdot f'(t_i, y_i) + \dots + \frac{h^n}{n!} \cdot f^{[n-1]}(t_i, y_i), \quad (3)$$

where h is the size of integration step, $y_i = y(t_i)$ is the previous value and $y_{i+1} = y(t_i + h)$ is the next value of the function $y(t)$ [5].

MTSM very effectively implements the variable-step-size, variable-order numerical calculation of differential equations using the Taylor series [2]. It is based on a recurrent calculation of the Taylor series terms for each integration step. Therefore, the complicated calculation of higher-order derivatives does not need to be performed, but rather the value of each Taylor series term can be numerically calculated [2]. Equation (3) can then be rewritten in the form

$$y_{i+1} = DY_0 + DY_1 + DY_2 + \dots + DY_n, \quad (4)$$

where DY denotes the Taylor series terms. MTSM transforms the input problem into a system of autonomous ODEs, which allows the recurrent calculation of terms of the Taylor series.

An important part of the method is an automatic integration order setting, i.e. using as many Taylor series terms as the defined accuracy requires. Let us denote as ORD the function which changes during the computation and defines the number

of Taylor series terms used in the current integration step ($ORD_{i+1} = n$).

MTSM allows for computation with arbitrary accuracy and step-size if variable-precision arithmetic and higher order of the method is used.

The first implementation of MTSM was TKSL/386 [6]. Currently, MTSM has been implemented and tested in MATLAB [7], in C/C++ languages (FOS [8] and TKSL/C software [9]). Additionally, the method can be effectively implemented in hardware [10].

Several other implementations of the Taylor series method in a variable order and variable step context were presented by different authors. TIDES software [11] and TAYLOR [12], which includes detailed description of a variable-step-size version. Other implementations based on Taylor series include ATOMF [13], COSY INFINITY [14], and DAETS [15]. The variable step-size variable-order scheme is also described in [16], [17], and [18], where simulations on a parallel computer are shown. The approach based on an approximate formulation of the Taylor methods can be found in [19].

A. Linear Problems

For linear systems of ODEs ($\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{b}$), the equation (3) can be rewritten in matrix-vector notation as

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h(\mathbf{A}\mathbf{y}_i + \mathbf{b}) + \frac{h^2}{2!}\mathbf{A}(\mathbf{A}\mathbf{y}_i + \mathbf{b}) + \dots + \frac{h^n}{n!}\mathbf{A}^{(n-1)}(\mathbf{A}\mathbf{y}_i + \mathbf{b}), \quad (5)$$

where \mathbf{A} is the constant Jacobian matrix and \mathbf{b} is the constant right-hand side. Moreover, Taylor series terms DY (4) can be computed recurrently using

$$\begin{aligned} DY_0 &= \mathbf{y}_i, \quad DY_1 = h(\mathbf{A}\mathbf{y}_i + \mathbf{b}), \\ DY_j &= \frac{h}{j}\mathbf{A}DY_{j-1}, \quad j = 2, \dots, n. \end{aligned} \quad (6)$$

B. Nonlinear Problems

The effective solution of nonlinear systems of ODEs is described. For such a system, Taylor series based numerical method can be implemented in a very effective way.

Equation (1) for nonlinear systems of ODEs can be rewritten as

$$\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{B}_1\mathbf{y}_{jk} + \mathbf{B}_2\mathbf{y}_{jkl} + \dots + \mathbf{b}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad (7)$$

where $\mathbf{A} \in \mathbb{R}^{ne \times ne}$ is the matrix for linear part of the system and $\mathbf{B}_1 \in \mathbb{R}^{ne \times nm_{jk}}$, $\mathbf{B}_2 \in \mathbb{R}^{ne \times nm_{jkl}}$ are the matrices for nonlinear part of the system, $\mathbf{b} \in \mathbb{R}^{ne}$ is the right-hand side for the forces incoming to the system, \mathbf{y}_0 is a vector of initial conditions, the symbol ne stands for the number of equations of the system of ODEs and the symbols nm_{jk} , nm_{jkl} mean the number of multiplications. The unknown function $\mathbf{y}_{jk} \in \mathbb{R}^{nm_{jk}}$ represents the vector of multiplications $\mathbf{y}_j \cdot \mathbf{y}_k$ and similarly $\mathbf{y}_{jkl} \in \mathbb{R}^{nm_{jkl}}$ represent the vector of multiplications $\mathbf{y}_{jj} \cdot \mathbf{y}_{kk} \cdot \mathbf{y}_{ll}$, where indexes $j, k, jj, kk, ll \in (1, \dots, ne)$ come from multiplications terms in the system (7). The operation ‘ \cdot ’ stands for

element-by-element multiplication, i.e. $\mathbf{y}_j \cdot \mathbf{y}_k$ is a vector $(y_{j1}y_{k1}, y_{j2}y_{k2}, \dots, y_{jn}y_{kn})^T$.

For simplification, the matrices \mathbf{A} , \mathbf{B}_1 , \mathbf{B}_2 and the vector \mathbf{b} are constant.

The higher derivatives of the terms $\mathbf{B}_1\mathbf{y}_{jk}$, $\mathbf{B}_2\mathbf{y}_{jkl}$ in (7) can be included in a recurrent calculation of Taylor series terms DY_{B1} and DY_{B2}

$$\begin{aligned} DY_{B1} &= h(\mathbf{B}_1\mathbf{y}_{jk}), \quad DY_{B2} = h(\mathbf{B}_2\mathbf{y}_{jkl}), \\ DY_{B1r} &= \frac{h}{r} \left(\mathbf{B}_1 \sum_{p=1}^r DY_{j,r-p} \cdot DY_{k,p-1} \right), \quad (8) \\ DY_{B2r} &= \frac{h}{r} \mathbf{B}_2 \sum_{q=0}^{r-1} DY_{jj,q} \cdot \left(\sum_{p=1}^r DY_{kk,r-p-q} \cdot DY_{ll,p-1} \right), \\ DY_r &= DY_{A_{r-1}} + DY_{B1_{r-1}} + DY_{B2_{r-1}}, \quad r = 2, \dots, n, \end{aligned}$$

where the linear term $DY_{A_{r-1}}$ is computed using (6).

III. SELECTED EXAMPLES

In this section, several examples of linear and nonlinear problems are presented and solved using the MTSM. The MTSM was implemented in MATLAB software [7] using vectorization. The results are compared with state-of-the-art non-stiff solvers in MATLAB, specifically ode23, ode45 and ode113 [20].

The main focus of the article are nonlinear problems to show the current development of the method, the linear ones were extensively analyzed in the previously published articles (e.g. [21], [22], [3]). For all numerical experiments, the error per step $eps = 10^{-12}$ and the maximal order of the MTSM is $maxORD = 60$, results are in Tables I–IV. All experiments were performed using MATLAB 2018b. Note that columns labeled as *Time of calculation* and *Ratio* are taken as a median value from 100 runs. Ratios of computation times $ratio = \mathbf{ode}/\mathbf{MTSM} \gg 1$ indicate significantly faster computation using MTSM.

A. Movement of the charged particle

Movement of the charged particle in the electromagnetic and electrostatic field is given by Lorenz (electromagnetic) force [23], [2]

$$\mathbf{f} = q(\mathbf{e} + \mathbf{v}\mathbf{b}_m), \quad (9)$$

where $\mathbf{e} = (e_x, e_y, e_z)^T$ is the external electric field, $\mathbf{b}_m = (b_{mx}, b_{my}, b_{mz})^T$ is the external magnetic field, vector \mathbf{v} is instantaneous velocity of the charged particle with charge q . The force vector \mathbf{f} can be substituted as $m\mathbf{a}$,

$$m\mathbf{a} = q(\mathbf{e} + \mathbf{v}\mathbf{b}_m). \quad (10)$$

We can further suppose, that acceleration a in x , y and z axis (x'' , y'' , z'') can be calculated as

$$\begin{aligned} x'' &= \frac{q}{m} (e_x + y'b_{mz} - z'b_{my}), \\ y'' &= \frac{q}{m} (e_y + z'b_{mx} - x'b_{mz}), \\ z'' &= \frac{q}{m} (e_z + x'b_{my} - y'b_{mx}). \end{aligned} \quad (11)$$

The problem can be transformed into a system of ODEs in a matrix-vector representation ($\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{b}$), where

$$\mathbf{A} = \begin{pmatrix} 0 & \frac{q}{m}b_z & -\frac{q}{m}b_y & 0 & 0 & 0 \\ -\frac{q}{m}b_z & 0 & \frac{q}{m}b_x & 0 & 0 & 0 \\ \frac{q}{m}b_y & -\frac{q}{m}b_x & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} x' \\ y' \\ z' \\ x \\ y \\ z \end{pmatrix}, \quad (12)$$

initial conditions are $\mathbf{y}_0 = (-8 \cdot 10^7, 0, 0, 0, 0, 0)^T$. Constants are set to $m = 9.10938356 \cdot 10^{-31}$ [kg], $q = -1.6 \cdot 10^{-19}$ [C], $b_{mx} = b_{my} = 0, b_{mz} = 0.5$ [T] and $e_x = e_y = e_z = 0$ [T]. Further, right-hand side $\mathbf{b} = (\frac{q}{m}e_x, \frac{q}{m}e_y, \frac{q}{m}e_z, 0, 0, 0)^T = \mathbf{0}$. The behavior of the electron is in Figure 1.

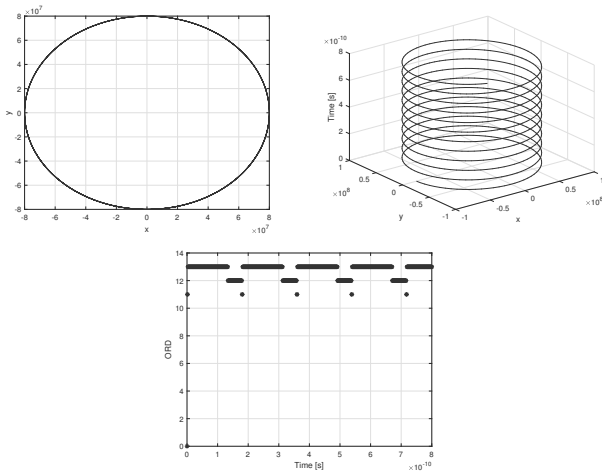


Figure 1. Position of the particle (left), trajectory of the particle (right), ORD function for $h = 10^{-12}$ (middle)

Results for the used parameters of the electromagnetic and electrostatic fields are in Table I. The MTSM calculates the problem faster, the average order of the method is 15.

Table I
RESULTS OF CALCULATIONS FOR MOVEMENT OF THE PARTICLE

Solver	Time of calculation [s]	Ratio
MTSM	0.0128435	–
ode23	1.81398	141.2
ode45	0.0289329	2.3
ode113	0.0272756	2.3

B. Simple pendulum

The simple pendulum is an idealization of a real pendulum. It consists of a point mass m , attached to an infinitely light rigid rod of length l that is itself attached to a friction-less pivot point. If displaced from its ideal vertical equilibrium position, this idealized pendulum will oscillate with a constant amplitude forever. There is no damping of the motion from friction at the pivot point or from air molecules impinging on the rod. Newton's second law provides the equation of motion:

$$l\phi'' = -g \sin(\phi), \quad (13)$$

where ϕ is the angular displacement of the pendulum from the vertical position, l is the length of the rod and g is the gravitational acceleration 9.81 [m s^{-2}] [24]. To slightly increase the complexity of a problem, the damping can be introduced. Equation (13) can be amended as follows:

$$ml\phi'' = -d\phi' - mg \sin(\phi), \quad (14)$$

where d represents the damping and m represents the mass of the pivot point (Figure 2).

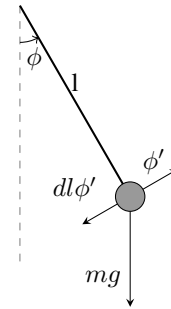


Figure 2. Simple pendulum with damping

Equation (14) can be simplified

$$\phi'' = -\frac{d}{m}\phi' - \frac{g}{l}\sin(\phi), \quad (15)$$

and linearized ($\sin(\phi) \approx \phi$) for small angles:

$$\phi'' = -\frac{d}{m}\phi' - \frac{g}{l}\phi. \quad (16)$$

Using (16), the linear IVP can be described by a matrix-vector representation

$$\mathbf{A} = \begin{pmatrix} -\frac{d}{m} & -\frac{g}{l} \\ 1 & 0 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \phi' \\ \phi \end{pmatrix} \quad (17)$$

where ϕ'_0 is the initial velocity and ϕ_0 is the initial position of the pendulum. Initial conditions of the linear problem are $\mathbf{y}_0 = (1, 0)^T$.

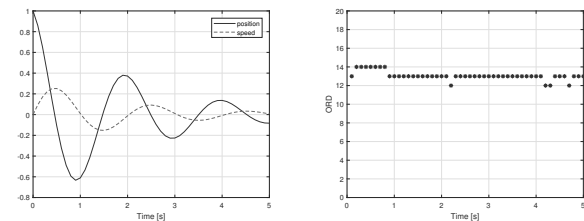


Figure 3. Linearized pendulum (left), ORD function for $h = 0.1$ (right)

Results for linearized pendulum are in Table II. The MTSM solves the linearized pendulum problem faster than the state-of-the-art MATLAB solvers. The average order of the method is 13.

Table II
RESULTS OF CALCULATIONS FOR LINEARIZED SIMPLE PENDULUM

Solver	Time of calculation [s]	Ratio
MTSM	0.00124377	–
ode23	0.294233	236.6
ode45	0.00512543	4.1
ode113	0.0102121	8.2

The problem becomes nonlinear when (15) is used. MTSM replaces $\sin(\phi)$ and $\cos(\phi)$ functions with auxiliary system of ODEs

$$\begin{aligned} y_3' &= \phi' \cos(\phi) = y_1 y_4 & y_3(0) &= \sin(\phi), \\ y_4' &= -\phi' \sin(\phi) = y_1 y_3 & y_4(0) &= \cos(\phi). \end{aligned} \quad (18)$$

The matrix-vector representation of the IVP (7) follows

$$\mathbf{A} = \begin{pmatrix} -\frac{d}{m} & 0 & -\frac{g}{l} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \phi' \\ \phi \\ \sin(\phi) \\ \cos(\phi) \end{pmatrix}, \quad (19)$$

$$\mathbf{B}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & -1 \end{pmatrix}, \mathbf{y}_{jk} = \begin{pmatrix} y_1 y_4 \\ y_1 y_3 \end{pmatrix},$$

right-hand side \mathbf{b} is set to $\mathbf{0}$, initial conditions for the nonlinear problem are $\mathbf{y}_0 = (1, 0, \sin(0), \cos(0))^T$. Indices in vector \mathbf{y}_{jk} correspond to the indices used in multiplications in (18). The MTSM uses averagely 14 terms of the Taylor series (see Figure 4).

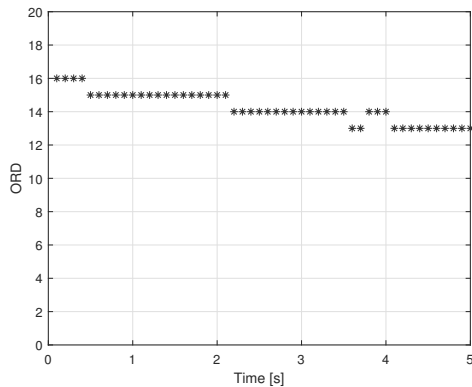


Figure 4. ORD function $h = 0.1$ (right)

Results for nonlinearized pendulum are in Table III.

Table III
RESULTS OF CALCULATIONS FOR NONLINEARIZED SIMPLE PENDULUM

Solver	Time of calculation [s]	Ratio
MTSM	0.00484634	–
ode23	0.292521	60.4
ode45	0.00448965	0.9
ode113	0.0113366	2.3

When comparing the results of linear and nonlinear approaches, note that the MTSM solver is faster or comparable to the state-of-the-art MATLAB solvers.

C. Fourier coefficients

Fourier series are the limit of the sequence of trigonometric polynomials that have a cosine part and a sine part. They are mainly used in the study of phenomena with a periodic character. The advantage of the Fourier series is that the requirements for their convergence to the developed function are weaker than in the case of the Taylor series. For example, the existence of derivatives of all orders of a given function at a given point is not required. The calculation of coefficients can be (especially when using numerical methods) easier than for Taylor series. Any periodic function $f(t)$ can be written as a sum of harmonics (20) called Fourier series

$$f(t) = \frac{a_0}{2} + a_1 \cos(\omega t) + a_2 \cos(2\omega t) + \dots + b_1 \sin(\omega t) + b_2 \sin(2\omega t) + \dots, \quad (20)$$

where

$$a_0 = \frac{2}{T} \int_0^T f(t) dt, \quad (21)$$

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt, \quad k = 1, 2, 3, \dots, n, \quad (22)$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt, \quad k = 1, 2, 3, \dots, n. \quad (23)$$

The calculation of Fourier series coefficients requires the calculation of definite integral. The definite integral

$$Y = \int_0^T f(x) dx \quad (24)$$

can be transformed to the differential equation

$$y' = f(x) \quad y(0) = 0, \quad (25)$$

where $y(T) = Y$ and T denotes the maximum time. Note that the same approach can be used for the calculation of multiple definite integrals. More details about transformation and numerical solution of definite integrals using MTSM can be found in [25].

For example, the analytic calculation of coefficient A_2 using definite integral

$$A_2 = \frac{2}{T} \int_0^T f(t) \cos(2\omega t) dt \quad (26)$$

can be transformed to IVP

$$a_2' = \frac{2}{T} f(t) \cos(2\omega t), \quad a_2(0) = 0. \quad (27)$$

The solution of the IVP in the maximum time T represents the calculated value ($A_0 \approx a_0(T)$, $A_2 \approx a_2(T)$). In this article the function

$$f(t) = \sin^2(\omega t) \tag{28}$$

is used ($\omega = \frac{2\pi}{T}$ [rad s⁻¹], $T = 2$ [s]). Coefficients of the Fourier series for (28)

$$\sin^2(\omega t) = \frac{a_0}{2} + a_2 \cos(2\omega t) \tag{29}$$

can be calculated analytically

$$\sin^2(\omega t) = \frac{1}{2} - \frac{1}{2} \cos(2\omega t) \tag{30}$$

as $A_0 = 1$ and $A_2 = -\frac{1}{2}$. The coefficients can be calculated numerically using the system of ODEs

$$\begin{aligned} a_0' &= f(t) & a_0(0) &= 0 \\ a_2' &= \frac{2}{T} f(t) \cos(2\omega t) & a_2(0) &= 0. \end{aligned} \tag{31}$$

The matrix-vector representation of the nonlinear system (7) is in the form

$$\mathbf{A} = \begin{pmatrix} 0 & \omega & 0 & 0 & 0 & 0 \\ -\omega & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & -2\omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \sin(\omega t) \\ \cos(\omega t) \\ \sin(2\omega t) \\ \cos(2\omega t) \\ a_0 \\ a_2 \end{pmatrix},$$

$$\mathbf{B}_1 = (0, 0, 0, 0, 1, 0)^T, \mathbf{B}_2 = (0, 0, 0, 0, 0, \frac{2}{T})^T,$$

$$\mathbf{y}_{jk} = \sin(\omega t) \sin(\omega t) = y_1 y_1,$$

$$\mathbf{y}_{ijk} = \sin(\omega t) \sin(\omega t) \cos(2\omega t) = y_1 y_1 y_4, \tag{32}$$

then the right-hand side \mathbf{b} is set to $\mathbf{0}$ and initial conditions are $\mathbf{y}_0 = (0, 1, 0, 1, 0, 0)^T$.

For $T = 2$ [s], the final values of curves in Figure 5 show the calculated values of coefficients a_0 and a_2 .

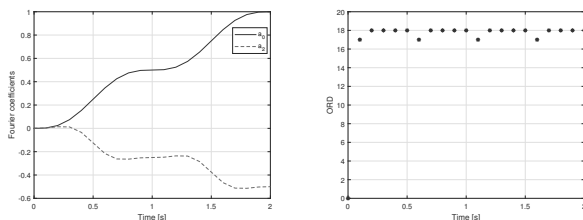


Figure 5. Fourier coefficients of $\sin^2(\omega t)$ (left), ORD function for $h = 0.1$ (right)

Results of the calculation of Fourier coefficients are in Table III. The column labeled *Error* shows the mean absolute error between analytic (A_0, A_2) and numerical solution (a_0, a_2).

Table IV
RESULTS OF CALCULATIONS FOR FOURIER COEFFICIENTS

Solver	Time of calculation [s]	Ratio	Error
MTSM	0.00510834	-	3.54346e-15
ode23	0.443919	86.9	1.57951e-12
ode45	0.00728959	1.4	1.44005e-13
ode113	0.00961171	1.9	9.64044e-15

The results show that MTSM is faster and more accurate than all tested state-of-the-art solvers. The average order used by the method is 20. The calculated coefficients can be used to reconstruct the original function.

IV. CONCLUSION

This paper was dedicated to Jiří Kunovský and his amazing life long work in the area of numerical computations of differential equations. The paper dealt with the numerical solution of both linear and nonlinear IVPs represented by a system of ODEs. The results showed, that MTSM overperforms state-of-the-art MATLAB ode solvers in most cases. The future research will focus on further improvements of the nonlinear MTSM solver.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations excellence in science - LQ1602”. The paper also includes the results of the internal BUT FIT project FIT-S-17-4014.

REFERENCES

- [1] P. Sehnalová and V. Šátek, “35 years of taylor-kunovsky simulation language,” in *16th International Conference of Numerical Analysis and Applied Mathematics*, vol. 2116, no. 7. American Institute of Physics, 2019, pp. 1–4. [Online]. Available: <https://www.fit.vut.cz/research/publication/11799>
- [2] J. Kunovský, “Modern taylor series method,” Habilitation work, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, 1994.
- [3] F. Kocina, G. Nečasová, P. Veigend, V. Šátek, and J. Kunovský, “Parallel Solution of Higher Order Differential Equations,” in *14th International Conference on High Performance Computing & Simulation*. Institute of Electrical and Electronics Engineers, 2016, ISBN 978-1-5090-2088-1.
- [4] P. Veigend, V. Šátek, and G. Nečasová, “Model of the telegraph line and its numerical solution,” *Open Computer Science*, vol. 8, no. 1, pp. 10–17, 2018. [Online]. Available: <https://www.fit.vut.cz/research/publication/11666>
- [5] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I. vol. Nonstiff Problems*. Springer-Verlag Berlin Heidelberg, 1987, ISBN 3-540-56670-8.
- [6] J. Kunovský, *TKSL*. [Online]. Available: <https://www.fit.vut.cz/research/product/51/en>
- [7] *MATLAB and Simulink software*, 2017. [Online]. Available: <http://www.mathworks.com>
- [8] F. Kocina, *FOS WEBGUI*. [Online]. Available: http://www.fit.vutbr.cz/iveigend/fos/index_en.php
- [9] V. Šátek, *High Performance Computing Research Group*. [Online]. Available: <https://www.fit.vut.cz/research/group/hpcl/en>
- [10] F. Kocina, J. Kunovský, G. Nečasová, V. Šátek, and P. Veigend, “Parallel solution of higher order differential equations,” in *Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS 2016)*. Institute of Electrical and Electronics Engineers, 2016, pp. 302–309.
- [11] R. Barrio, M. Rodríguez, A. Abad, and F. Blesa, “TIDES: A free software based on the Taylor series method,” *Monografías de la Real Academia de Ciencias de Zaragoza*, vol. 35, pp. 83–95, 2011.

- [12] A. Jorba and M. Zou, "A software package for the numerical integration of ODE by means of high-order Taylor methods," in *Exp. Math.*, vol. 14, 2005, pp. 99–117.
- [13] Y. F. Chang and G. Corliss, "Atomf: solving odes and daes using Taylor series," *Computers Math. Applic.*, vol. 28, pp. 209–233, 1994.
- [14] M. Berz, "COSY INFINITY version 8 reference manual," *Technical Report MSUCL-1088, National Superconducting Cyclotron Lab., Michigan State University, East Lansing, Mich.*, 1997.
- [15] N. S. Nedialkov and J. Pryce, "Solving differential algebraic equations by Taylor series III. the DAETS code," *JNAIAM J. Numer. Anal. Ind. Appl. Math.*, vol. 3, pp. 61–80, 2008.
- [16] R. Barrio, F. Blesa, and M. Lara, "VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs," in *Computers and Mathematics with Applications*, vol. 50, 2005, pp. 93–111.
- [17] R. Barrio, "Performance of the Taylor series method for ODEs/DAEs," in *Applied Mathematics and Computation*, vol. 163, 2005, pp. 525–545, ISSN 00963003.
- [18] P. Mohazzabi and J. L. Becker, "Numerical Solution of Differential Equations by Direct Taylor Expansion," *Journal of Applied Mathematics and Physics*, vol. 05, no. 03, pp. 623–630, 2017. [Online]. Available: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jamp.2017.53053>
- [19] A. Baeza, S. Boscarino, P. Mulet, G. Russo, and D. Zorío, "Approximate Taylor methods for ODEs," *Computers & Fluids*, vol. 159, pp. 156 – 166, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045793017303584>
- [20] *MATLAB: Choose an ODE Solver*, 2017. [Online]. Available: <https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html>
- [21] P. Veigend, G. Nečasová, and V. Šátek, "Model of the telegraph line and its numerical solution," *Open Computer Science*, vol. 8, pp. 1–8, in press, 2018, <https://doi.org/10.1515/comp-2018-0002>. [Online]. Available: <https://www.degruyter.com/view/j/comp.2018.8.issue-1/issue-files/comp.2018.8.issue-1.xml>
- [22] F. Kocina, "Advanced electronic circuits simulation methods," Ph.D. thesis, Brno University of Technology, Faculty of Information Technology, 2017. [Online]. Available: <https://www.fit.vut.cz/study/phd-thesis/623/>
- [23] P. Richard, L. FEYNMAN, B. ROBERT, and M. SANDS, *FEYNMAN LECTURES ON PHYSICS: The New Millennium Edition*. Basic Books, 2015.
- [24] G. L. Baker and J. A. Blackburn, *The pendulum: a case study in physics*. Oxford University Press, 2005.
- [25] J. Chaloupka, F. Kocina, P. Veigend, G. Nečasová, V. Šátek, and J. Kunovský, "Multiple integral computations," in *14th International Conference of Numerical Analysis and Applied Mathematics*, no. 1863. American Institute of Physics, 2017, pp. 1–4. [Online]. Available: <https://www.fit.vut.cz/research/publication/11226>

A Simulation Study of Depot with Multiple Temperature Modes: Maximum Flow Estimation

Marie Nedvěďová

University of Pardubice

Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
marie.nedvedova@student.upce.cz

Alena Pozdílková

University of Pardubice

Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
alena.pozdilкова@upce.cz

Jaroslav Marek

University of Pardubice

Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
jaroslav.marek@upce.cz

Michael Bažant

University of Pardubice

Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
michael.bazant@upce.cz

Abstract— The aim of this paper is to make a truck loading simulation in a large area with several warehouses each with a given temperature mode of stored goods (frozen, refrigerated, and unrefrigerated goods). When loading goods in different temperature modes, time delays occur when trucks are crossing between ramps (ramps for various temperature modes are separated). The problem is that if the number of ramps is insufficient, queues may arise. The aim of the paper is to quantify the dependence of the loading time on the number of loaded pallets where we consider usually achievable loading area limits of trucks when planning multiple trips. We assume that customer order quantities are given by a Beta distribution. Knowledge of this density estimate allow us to assess the maximum flow through the depot. Arena software is used for the simulation.

Keywords—depot; flow of goods; minimizing queues; loading simulation

I. INTRODUCTION

In this article, we will analyze loading of goods at ramps from warehouses of goods that are stored in three temperature modes. Our study is closely related to the family of Vehicle routing problems (VRP), Bin packing problems (BPP) and Traveling salesman problems (TSP). The reader can get a survey of the methods used to solve VRP in articles (Angel [2]), (El-Sherbeny [4]), (Serap [7]), and in the book (Simchi-Levi [8]). In the literature devoted to BPP, VRP or TSP, the issue of setting the vehicle exit time is not usually addressed. Various problems have been formulated in connection with the loading of goods, see (Alonso [1]), (Calvet [3]). In (Le-Duc [5]) Order batching problem (OBP) is solved within a 2-block rectangular warehouse with the aim to obtain average throughput time of random order and to assess optimal picking batch size.

Contradictory to our assumption that frequency between of arrivals is constant, authors in (Le-Duc [5]) consider the Poisson distribution for order arrival to the system.

The objective of our research is to optimize warehouse productivity by maximizing the flow of goods in the depot and reducing the waiting time of trucks at ramps. Cf. (Stadt [9]). In our study, we will describe the flow in the depot through simulation software. Special attention will be paid to the time between line arrivals so that queues do not occur at ramps. Our study complements the VRP assignment when we need to estimate the maximum number of vehicles loaded simultaneously. Therefore, we will focus only on assessing the impact of distribution of lines with mixes of goods on the loading speed. The inputs for our analysis are ramp counts for warehouses with three different temperature modes, and statistical characteristics describing the number of pallets distributed to customers in these temperature modes. The issue of flow of goods in the depot is also examined from the aspect of heterogeneous vehicle fleet structures. However, we will not perform price analysis in this article. A guide to our analysis was provided by the book (Kelton [6]).

II. MIXES OF GOODS SAMPLE

A. Data

The data for our study will be prepared by simulation taking into account the knowledge we have gained from the information about the actual loading of goods from a certain logistics company. A linear function describing the loading time per vehicle will be available. Four different types of mixes of temperature modes of the loaded goods arise during loading. Each mix of goods has its own mean value of commodity size.

While basic characteristics of mixes of goods were shown in Tab. I, mix percentages are given by Fig. 1. It has been shown that the size of the goods to be distributed can be described by a Beta distribution. This knowledge will be used in simulation, see (Stewart [10]).

By the moment method we can estimate the parameters as follows:

$$\hat{\alpha} = \bar{x} \frac{\bar{x}(1 - \bar{x})}{s - 1},$$

$$\hat{\beta} = (1 - \bar{x}) \frac{\bar{x}(1 - \bar{x})}{s - 1}.$$

The Data preparation procedure will be described and shown in the upcoming section.

TABLE I. A SETTING OF GOODS FEATURE VALUES

	Temperature mode	Alpha	Beta	Avg. [pcs.]	s [pcs.]
Pure	Frozen	0.0908	0.4347	5.58	9.86
	Refrigerated	0.1942	0.5342	8.86	11.07
	Unrefrigerated	0.2145	0.2771	14.37	13.57
	Frozen Refrigerated	0.1603	0.2919	11.72	13.09
Mixes	Frozen Refrigerated	0.3835	2.4289	4.42	5.89
	Refrigerated Unrefrigerated	0.3904	1.5917	6.45	7.57
	Frozen Refrigerated Unrefrigerated	0.1182	0.1226	16.33	14.91
	Frozen Refrigerated Unrefrigerated				

^a Source: own.

Size of orders according to temperature regime.

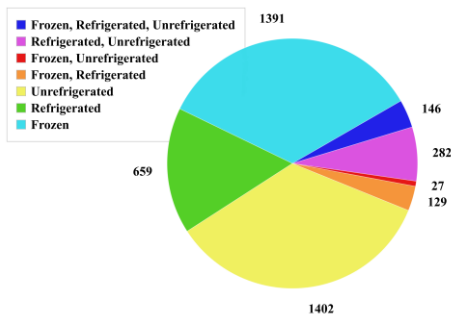


Fig. 1. Distribution of mixes. Source: own.

The most important input parameter is the loading time t of p pallets, which will be described by the function $t = 7 + p$. Other input parameters are listed in the following tables. Tables II, III, and IV outline the number of customers, vehicles, their capacities, and the number of ramps.

TABLE II. CAPACITIES IN DEPOTS

Depot	Ramps - frozen	Ramps - refrigerated	Ramps - unrefrigerated
Ramps	6	4	5
Hour capacity	198	134	165

^b Source: own

TABLE III. VEHICLE FLEET STRUCTURE

Vehicle	No. of cars	No. of pallets
Evaporator semitrailer	56	33
2 evaporators semitrailer	50	33
Lorry A	9	20
Lorry A2	17	18
Lorry B	12	15
Lorry C	6	8

TABLE IV. VARIANTS OF CUMULATION OF GOODS OF DIFFERENT TEMPERATURE REGIMES ON LINES

		V0	V1	V2	V3	V4
Pure	Frozen	34 %	20 %	16 %	12 %	4 %
	Refrigerated	17 %	10 %	8 %	6 %	2 %
	Unrefrigerated	34 %	20 %	16 %	12 %	4 %
	Sum of Pure	85 %	50 %	40 %	30 %	10 %
Mixes	Frozen Refrigerated	3 %	10 %	12 %	14 %	18 %
	Frozen Unrefrigerated	7 %	23 %	28 %	33 %	42 %
	Refrigerated Unrefrigerated	1 %	3 %	4 %	5 %	6 %
	Frozen Refrigerated Unrefrigerated	4 %	13 %	16 %	19 %	24 %
	Sum of Mixes	15 %	50 %	60 %	70 %	90 %

^c Source: own.

III. THE SIMULATION

A. The Design of Simulation

Firstly, we developed a structure of the simulation model. The focus of our simulation study is to find the optimal period for the arrival of vehicles to the depot. First of all, it is necessary to prepare input data. We need to set a strategy for selecting a vehicle from a heterogeneous fleet of vehicles. The vehicle is assigned one of four categories of mixes. Next, we simulate the goods for the selected mix with the characteristics from Tab. I.

The simulation steps are:

1) Allocation of orders from the same temperature regimes to vehicles (cf. last column of Tab. I):

a) Simulation of random number according Beta distribution with parameters from Tab. I for all temperature modes and combinations thereof.

b) Select a vehicle from the fleet of camions described in Tab. III.

c) Randomly assign a temperature mode with probabilities given in Tab. IV.

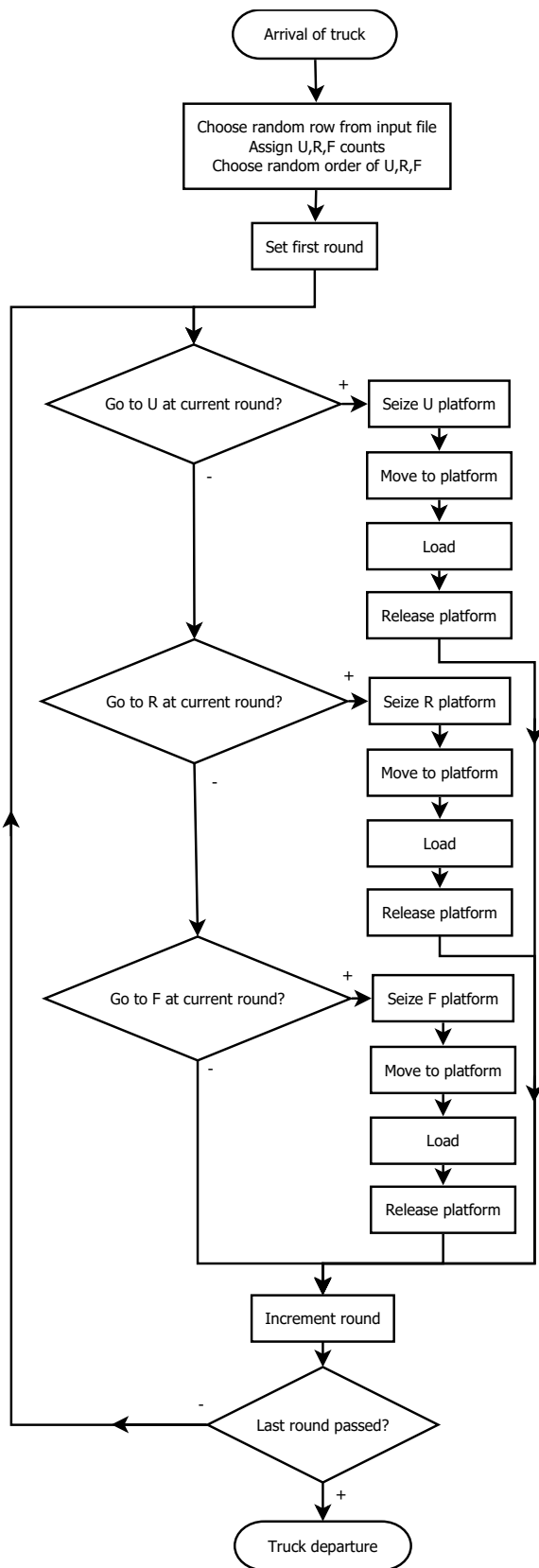


Fig. 2. Flow Chart. Source: own.

d) Packing of the selected vehicle to its capacity from sets of orders (in the temperature mode selected in point 3.), generated under point 1. If the size of the goods reaches 90 percent of the car's capacity, then stop adding more orders.

2) Variants of different proportions of goods lines in a single temperature regime, see Tab. IV:

a) Selection of already created lines in A) point 4 with probabilities from Tab. IV.

b) Creation of lines with mixes of goods with cumulative relative sum in 5 variants reaching values of 50, 60, 70, 80, and 90 percent.

3) Simulation of car arrival at ramps, cf. Tab. III:

a) The pause between car arrivals is set in the depot.

b) Based on the basic file, the type of truck and the type of goods (temperature regimes) it carries are generated.

c) The order in which the truck will go to each type of ramp is randomly selected.

d) The ramps are gradually occupied by the truck according to the assigned order. (the truck backs up to the ramp and loads the appropriate number of pallets of goods.)

e) After loading all goods, the truck leaves the depot.

The implementation of the presented simulation is shown in Fig. 2.

B. Simulation results

The results obtained by using Arena software are given in Tables V - IX. For various options in the tables we present load time values of the last pallet, average of simultaneously loaded cars, average wait-time for every ramp, and hourly flow for every temperature mode. The results are also presented graphically in Fig. 3 to 7.

TABLE V. DEPOT 1, VARIANT 0

Delay between arrivals[minutes]	4	5	6	7	8	9
Load time of a last pallet [hours]	5:54	6:12	7:21	8:43	9:57	11:03
Avg. of simultaneously loaded cars	9.73	7.77	6.49	5.56	4.87	4.32
Avg. waittime Frozen [minutes]	0:40	0:04	0:00	0:00	0:00	0:00
Avg. waittime Refrigerated [minutes]	1:01	0:32	0:18	0:14	0:11	0,07
Avg. waittime of Unrefrigerated [minutes]	3:00	0:32	0:07	0:00	0:00	0:00
Hourly flow Frozen	156.22	123.98	104.53	86.96	75.57	67.69
Hourly flow Refrigerated	83.91	68.55	55.77	47.96	42.27	38.32
Hourly flow Unrefrigerated	162.77	131.78	114.10	96.27	84.36	75.93

TABLE VI. DEPOT 1, VARIANT 1

Delay between arrivals [minutes]	4	5	6	7	8	9
Load time of a last pallet [hours]	5:08	6:19	7:28	8:48	10:02	11:21
Avg. of simultaneously loaded cars	11.63	9.56	7.98	6.84	5.97	5.31
Avg. waittime Frozen [minutes]	0:47	0:11	0:04	0:00	0:00	0:00
Avg. waittime Refrigerated [minutes]	1:59	0:43	0:14	0:11	0:07	0:04
Avg. waittime of Unrefrigerated [minutes]	14:17	2:24	0:32	0:11	0:04	0:00
Hourly flow Frozen	128.91	101.94	86.29	73.94	64.64	57.87
Hourly flow Refrigerated	84.10	69.44	58.02	48.04	42.91	38.54
Hourly flow Unrefrigerated	177.62	146.43	124.78	106.06	92.20	80.95

TABLE VII. DEPOT 1, VARIANT 2

Delay between arrivals [minutes]	4	5	6	7	8	9
Load time of a last pallet [hours]	4:55	6:08	7:21	8:38	9:51	11:03
Avg. of simultaneously loaded cars	11.65	9.51	7.89	6.75	5.90	5.24
Avg. waittime Frozen [minutes]	1:05	0:18	0:04	0:00	0:00	0:00
Avg. waittime Refrigerated [minutes]	2:46	0:43	0:18	0:11	0:04	0:00
Avg. waittime of Unrefrigerated [minutes]	10:52	1:48	0:32	0:07	0:04	0:00
Hourly flow Frozen	133.31	107.99	90.05	76.47	67.80	61.56
Hourly flow Refrigerated	100.57	79.10	66.14	55.20	47.40	42.21
Hourly flow Unrefrigerated	174.51	140.64	116.79	100.79	88.50	78.37

TABLE VIII. DEPOT 1, VARIANT 3

Delay between arrivals [minutes]	4	5	6	7	8	9
Load time of a last pallet [hours]	5:42	6:18	7:29	8:49	10:01	11:20
Avg. of simultaneously loaded cars	11.06	9.05	7.57	6.50	5.68	5.05
Avg. waittime Frozen [minutes]	0:47	0:07	0:04	0:00	0:00	0:00
Avg. waittime Refrigerated [minutes]	2:42	0:54	0:32	0:14	0:04	0:04
Avg. waittime of Unrefrigerated [minutes]	12:36	2:28	0:43	0:14	0:04	0:00
Hourly flow Frozen	111.74	99.94	83.95	71.36	62.55	56.28
Hourly flow Refrigerated	74.83	65.26	54.83	46.58	42.02	36.89
Hourly flow Unrefrigerated	165.50	153.88	128.82	110.23	95.90	83.96

TABLE IX. DEPOT 1, VARIANT 4

Delay between arrivals [minutes]	4	5	6	7	8	9
Load time of a last pallet [hours]	4:59	6:15	7:30	8:57	10:06	11:22
Avg. of simultaneously loaded cars	11.03	8.84	7.37	6.31	5.51	4.90
Avg. waittime Frozen [minutes]	0:29	0:07	0:00	0:00	0:00	0:00
Avg. waittime Refrigerated [minutes]	1:41	0:32	0:14	0:07	0:04	0:00
Avg. waittime of Unrefrigerated [minutes]	6:07	1:05	0:14	0:04	0:00	0:00
Hourly flow Frozen	138.79	110.96	90.87	76.23	67.23	61.13
Hourly flow Refrigerated	93.41	71.49	61.30	50.16	43.90	38.99
Hourly flow Unrefrigerated	170.34	138.10	115.31	98.36	87.66	76.27

The simulated results showed that the highest hourly flow occurs at the frequency of exits being one per 4 minutes. However, the estimated hourly flows exceed the maximum allowable flow of 165 pallets for unrefrigerated goods. With an exit frequency of 4 minutes, the large hourly flow rate and the shortest loading time of the last pallet are redeemed by the longest waiting time of up to 5 minutes.

Therefore, it seems preferable to set the simulation to 5 minutes, where the waiting of the vehicles in front of the ramps was approximately 1 minute. With a 5-minute frequency selection, it appeared that the local maximum of total hourly flow occurred in Variant 2 (70 percent of mixes).

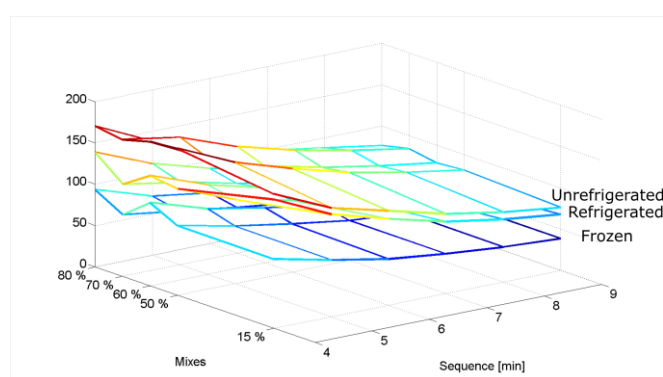


Fig. 3. Hourly flow – Unrefrigerated, Refrigerated, Frozen. Source: own.

IV. CONCLUDING REMARKS

Our goal was to prepare the basic pillars for the possibility of building an algorithm for VRP solution, which will guarantee loading without queues. Use of the simulation software brought interesting and fruitful results. We found out that there will be no queues on the ramps if the lines are serviced with a period of 6 minutes.

ACKNOWLEDGMENT

This research was supported by the project Technology development for intelligent traffic flow, Ministry of Industry and Trade, Czech Republic.

References

- [1] M. T. Alonso, R. Alvarez-Valdes, F. Parreño, and J. M. Tamarit, "Algorithms for pallet building and truck loading in an interdepot transportation problem," *Mathematical Problems in Engineering*, vol. 2016, Article ID 3264214, 11 pages, 2016.
- [2] A. Juan Angel, E. Grasman Scott, J. Faulin, D. Riera, C. A. Méndez, and B. Ruiz, "Applying simulation and reliability to vehicle routing problems with stochastic demands." *Proceedings of the 16th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, RCRA@AI*IA 2009, Reggio Emilia, Italy, December 11-12, 2009.*
- [3] L. Calvet, D., Wang, A. Juan, and L. Bové, "Solving the multidepot vehicle routing problem with limited depot capacity and stochastic demands." *International Transactions in Operational Research*, vol. 26, no. 2, pp. 458-484, 2019.
- [4] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods". *Journal of King Saud University*, Volume 22, pp. 123-131, 2010.
- [5] T. Le-Duc, and R. De Koster, "Travel time estimation and order batching in a 2-block warehouse." *European Journal of Operational Research*, vol. 176, no. 1, pp. 374-388, 2007.
- [6] W. D. Kelton, P. Randal, R. Sadowski, and N. B. Zupick, "Simulation with Arena." Sixth edition. New York, N.Y: McGraw-Hill Education, 2015.
- [7] E.C. Serap, R.Y. Harun, S. İrem, and S. Hanife, "A new approach for solution of vehicle routing problem with hard time window: an application in a supermarket chain". *Sadhana*, vol. 42, no. 12, pp. 2067-2080, 2017.
- [8] D. Simchi-Levi, X. Chen, and J. Bramel, "The logic of logistics: theory, algorithms, and applications for logistics management (Springer series in operations research and financial engineering)." Prentice-Hall, Inc., 2013.
- [9] F. H. Staudt, M. di Mascolo, G. B. Alpan, and C.M.T. Rodriguez, "Warehouse performance measurement: classification and mathematical expressions of indicators". In *ILS 2014 - 5th International Conference Information Systems, Logistics and Supply Chain. Dinalog Beta Trail, Breda, Netherlands*, pp. 1-9, Aug. 2014.
- [10] W. J. Stewart, "Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling." Princeton, Princeton University Press, 2009.

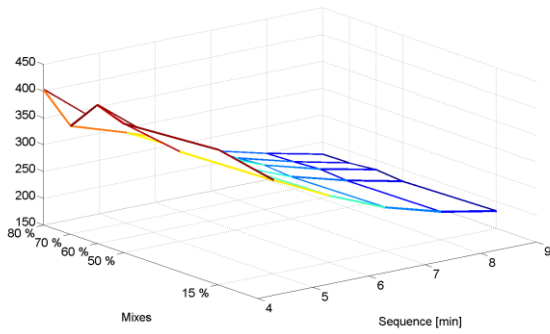


Fig. 4. Sum of hourly flow. Source: own.

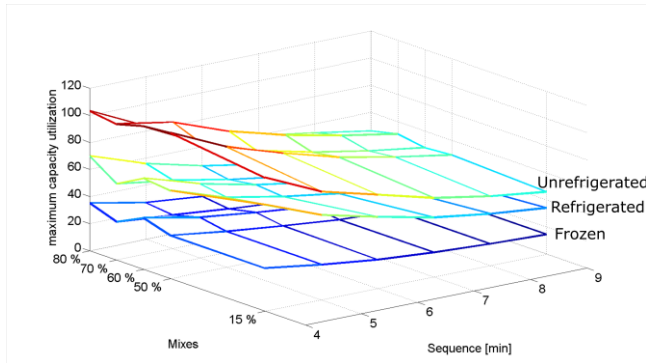


Fig. 5. Percentage usage of the ramp capacity. Source: own.

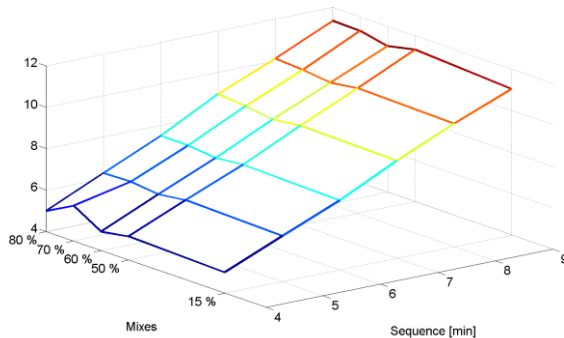


Fig. 6. Loading time of the last pallet [in hours]. Source: own.

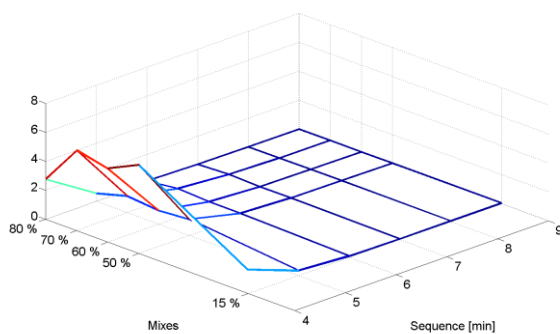


Fig. 7. Average waiting time of the car on all three ramps [in hours]. Source: own.

HaskellCompass: Extending the CodeCompass comprehension framework for Haskell

1st Boldizsár Németh

Department of Programming Languages and Compilers
Eötvös Loránd University
Budapest, Hungary
nboldi@elte.hu

2nd Tibor Brunner

Department of Programming Languages and Compilers
Eötvös Loránd University
Budapest, Hungary
bruntib@caesar.elte.hu

Abstract—While functional programming is an emerging and efficient way to express complex software, the tool support for functional programming languages, like Haskell, is not always satisfactory. There are no tools that can provide code insight for large Haskell projects that limits the potential of the language’s usability.

CodeCompass is an open-source LLVM/Clang-based tool developed by Ericsson Ltd. and Eötvös Loránd University, Budapest to help the understanding of large legacy software systems. Based on the LLVM/Clang compiler infrastructure, CodeCompass gives exact information on complex C/C++ language elements and is regularly used by hundreds of designers and developers.

In this paper we present a Haskell plugin for the CodeCompass tool that is able to analyse large Haskell projects and provide useful information for the developers. We suggest useful comprehension queries and actions that enhance developer efficiency. Additionally we show the implementation of such a solution and measure the performance cost of using it.

Index Terms—Code comprehension, CodeCompass, Functional programming, Haskell

I. INTRODUCTION

CodeCompass is a pluginable, extensible, open source code comprehension framework [1]. Its main purpose is to help understanding large code bases by gathering static analysis information from the whole knowledge portfolio and to present it to the user through various navigation and visualization techniques.

The comprehension setup has two phases. First the source code has to be parsed to collect information and store them into a database. In the second phase a server presents the collected information to the clients on a public API. An example web-based GUI client is a standard web browser which queries the server from JavaScript, however it is also possible to create other clients for the users’ own purposes, for example, editor plug-ins [2].

Code comprehension has been well studied in relation to imperative and object-oriented programming languages, but is not exhaustively explored for functional programming languages.

Some of the concepts are similar in imperative and functional programming languages. For example, call chains and

dependence relations between modules have their functional counterparts. Some concepts, like inheritance of implementation (class inheritance) are absent from functional programming languages, and others, like inheritance of specification (inheritance of interface) can be found, however in different form.

In this paper we describe a method to extract static data about functional programs and store them in the CodeCompass system to provide code comprehension tools for Haskell developers. We implemented this method as HaskellCompass, a Haskell plugin for the CodeCompass framework.

For the design of HaskellCompass, we used the GHC compiler to provide complete language support as well as support for the GHC-specific language extensions. We wrote a compiler plugin that operates on GHC’s program representation (a source plugin [3]).

GHC uses different representations during the compilation process. Different representations contain different data about the program being compiled. The HaskellCompass analyzer sets up hooks in each of the compilation steps using the plugin system. (See Figure 1.) This way it can analyze each of the representations of the program and collect all data needed for the providing code comprehension for the project.

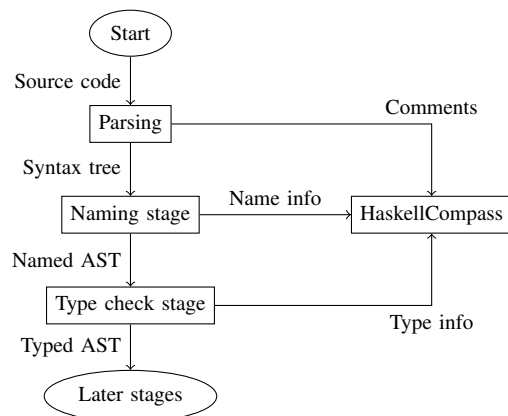


Fig. 1. HaskellCompass’s architecture

To record data for CodeCompass we need to perform the same compiler commands that were executed when the project

Supported by the European Union, co-financed by the European Social Fund, EFOP

was built, modified for recording data. Since GHC’s plugins can be used by simply giving flags to the compiler as command line arguments, it is easy to integrate HaskellCompass into any build environment. Moreover, CodeCompass provide tools to record and replay compiler calls, so even if the build system could not be manipulated there is a way to analyze the source code.

The paper is organised as follows:

Section II describes the related tools and methods already existing in the Haskell ecosystem. Section III presents an overview of the CodeCompass system that HaskellCompass is a part of. Section IV gives a basic set of code comprehension actions that we designed to help Haskell programmers. Section V describes the high-level design of the HaskellCompass system. Section VI shows some of the implementation details of the HaskellCompass system. Section VII evaluates the performance of the HaskellCompass system on a large project.

II. RELATED WORK

The most popular Haskell compiler, GHC, is now over 18 years old, and has been under continuous active development since its inception. Today, GHC releases are downloaded by hundreds of thousands of people, the online repository of Haskell libraries has over 7000 packages, GHC is used to teach Haskell in many undergraduate courses, and there are a number of instances of Haskell being depended upon commercially [4].

The tooling support for Haskell have not exited its experimental stage. To this day, the majority of Haskell programmers are using text editors for editing Haskell source code. There have been several approaches for a complete development environment for Haskell [5] [6] [7], but none of them became widespread.

Haddock, a tool for automatically generating documentation from Haskell source code [8] is useful for the understanding of API (application programming interface) documentation.

For recording the runtime behavior of Haskell programs, we have HPC, a toolkit to record and display Haskell Program Coverage. HPC includes tools that instrument Haskell programs to record program coverage, run instrumented programs, and display information derived from coverage data in various ways [9].

The tool support for profiling Haskell programs is also sophisticated, including parallel Haskell [10].

III. CODECOMPASS

Writing and reading a source code are two fundamentally different activities in software development. For example during the development we are editing only a few files at the same time which are more or less on the same abstraction level. When we are trying to comprehend the operation of a code snippet, we intend to browse its environment even on the higher (function, class, module, package, etc.) levels. Most of the times, development is text-based as opposed to code comprehension where visualization is helpful. During the development we are using code editor functionalities like code

completion, and in comprehension we require fast navigation among the references of a symbol. There are a lot of text editors and integrated development environments to support the former use case.

CodeCompass is a framework to support code comprehension activities. The goal of this tool is to gather all statically known information about the source code and present it to the developers. The extensibility of CodeCompass is due to its plug-in architecture. Besides language parsers this tool presents software metrics, fuzzy text search, symbol reference search, version control information, architecture models, call graphs, UML diagrams, pointer analysis, etc.

Code comprehension tools may handle the source code on different levels. The advantage of text-based tools like OpenGrok is that it is fast and easy to install them, however, their precision is limited. It is possible to distinguish classes, variables or functions, but it is not possible to find the exact definition of a function by selecting one of its usages on the user interface. Lacking overload resolution, text-based search is only suitable for enumerating all functions with the same qualified name. In contrast, CodeCompass chooses to parse the source code with the help of a compiler. This enables to resolve named entities and their relations. For example in case of an object-oriented language we have precise information about inheritance relationships among classes or in case of a functional language we know exactly what functions can be reached from a specific function along function calls. Having access to the type system of a statically typed language we have accurate data about the symbols and expressions.

In the further sections of this paper we present how Haskell language can be integrated into CodeCompass.

IV. CODE COMPREHENSION FOR HASKELL

In this section we analyze some of the code comprehension actions we thought of being useful for programming in Haskell. Most of them have counterparts in other programming languages, in these cases we analyze how they can be suited for Haskell.

A. Jump to Definition

Jumping to the definition of a symbol is one of the most simple code comprehension actions. It useful for any programming language, regardless of programming paradigm.

This feature is usually found in editor software even if program analysis is not implemented, but in those cases this works using a simple textual search.

1) *Motivation*: The user wants to check the definition of a function or value.

2) *Implementation*: When collecting the locations of the definitions we relied on GHC’s name resolution system. When encountering a name usage, we looked up the location of the definition of the name that was conveniently stored. Fortunately names are stored in GHC similarly whether they refer to values, functions, parameters, constructors or types, so it is relatively simple to extract their locations. Since in GHC’s representation module names are different from the names of

other program elements they have to be collected separately and connected to the referred modules.

B. Show Documentation

Hovering over a name should show the documentation for that element. This is also a universal code comprehension action and is similar to the *Jump to Definition* comprehension, but it does not move the focus of the user to the definition, but shows only the documentation.

1) *Motivation*: The user learns the information that is recorded by the author of a function in the definition call, without leaving the use site.

2) *Implementation*: API documentation is stored in special API comments, called Haddock comments in Haskell (Haddock is the tool to generate the documentation from these comments). Our plugin need to parse these comments to store the information in the database. Since Haddock's comment language is extensive, the information should be presented using Haddock's functionality to transform the comments into HTML code.

C. Show Type of Definition

Haskell has a very sophisticated type system. This prevents some programming mistakes but also makes it harder to understand the exact type structure of the program code.

The comprehension shows two types for a given definition, the *defined-type* is the more generic type of the definition that is determined by the site of definition. The other type is the *inferred-type* that is the type of the expression of the usage of the definition and is calculated by the type inference algorithm (see Figure 2). The defined-type is always more generic than the inferred-type, but for most of the times they are the same.

```
a :: Num a => a
a = 12 -- the definition of "a" has
      -- the defined-type "Num a => a"

b :: Int
b = a -- the expression "a" here has
      -- the inferred-type "Int"
```

Fig. 2. The defined and inferred type of a name

1) *Motivation*: The user checks the type of a definition to find out if it can be used in another context.

2) *Implementation*: The information regarding the types of definitions is produced in the type-checking stage of the compilation. When we extract the type information we use it to update the database and attach the type information to the names already present. We use location information to identify the names that are connected with the types that are found. The inferred-type is produced by applying the stored type arguments to the more generic type of the defined-type. The types of expressions and patterns can be easily accessed because they are subject to type-checking, but the type of types (called kinds in Haskell) cannot be simply queried, because they are not recorded when they are checked. Accessing this information needs future changes in the compiler.

D. List References

The list occurrences comprehension is the counterpart of *Jump to Definition*. While the latter goes from the usage of a name to the definition of the name, the former points from the definition of the name to all its usages.

List Usages is not as universal as *Jump to Definition*, because for this comprehension the system needs to be able to discover the parts of the program that use the given definition. This means that the system needs to use the build toolchain for discovering the modules. Fortunately this is already solved by the combination of the CodeCompass framework and using GHC plugins.

Since usages of a name can be classified into separate categories, *List References* can be expanded to only search for usages that fall into one of the categories. For example, if we are searching for the uses of a type class we can distinguish between uses that define instances of that type class and uses that use the type class as a predicate.

1) *Motivation*: The user wants to find out where a given function is used to check which parts of the program would be affected by a change.

2) *Implementation*: This comprehension uses the same database records as *Jump to Definition*, but navigates the connection in the opposite direction.

E. Jump to Instance Implementation

In Haskell a type class is a tool to create polymorphism. The type class defines an interface and the instances of the type class define the implementation of that interface for each type.

When using the *Jump to Definition* comprehension on a function that belongs to a type class, the system jumps to the definition of the type class where the function is defined. But in most cases, the user wants to check the implementation of the function that is called. In this cases the user is actually interested in the instance definition. See Figure 3 for an example.

GHC usually calculates which instance of a type class is called by a function call. By storing this information it is possible to jump to the location where the actual implementation of the called function is defined.

```
class Default a where
  default :: a

instance Default Int where
  default = 0

x :: (String, Int)
x = ("abc", default)
```

Fig. 3. The difference between Go to Definition and Go to Instance Implementation code comprehensions

The information is only present if the concrete types can be calculated by GHC's type checker. If the information cannot be found the comprehension should give the same results as *Jump to Definition* as a backup.

1) *Motivation*: The user explores the behavior of a polymorph function call.

2) *Implementation*: The implicit type arguments are found in GHC’s representation. Using this, the instance identifier can be located in one of GHC’s administrative tables. The connection between them is stored in the database.

F. List Uses of Instance Implementation

The comprehension *List Uses of Instance Implementation* is the counterpart of *Jump to Instance Implementation*. With this comprehension the user can search for the uses of a given instance implementation.

1) *Motivation*: This can help the user to find out which sections of the code are affected by a change to the implementation.

2) *Implementation*: This comprehension uses the same database records as *Jump to Instance Implementation*, but navigates the connection in the opposite direction.

G. Show Used Modules

The comprehension *Show Used Modules* shows the transitive closure of the modules that are imported by the current module.

This comprehension is only possible in languages where the usage of other modules is determined in compilation time.

1) *Motivation*: The user wants to find out which modules are required for a certain module to work. It is useful when the module needs to be separated or moved.

2) *Implementation*: The relationships between modules can be accessed after parsing the module. From this information the whole network of modules can be reconstructed. The source files, that are part of the project, are analyzed and connected to the corresponding module records. The hs-boot files that are used as an interface to the modules are ignored for simplicity.

H. Show Dependent Modules

The comprehension *Show Dependent Modules* shows the transitive closure of the modules that import the current module.

1) *Motivation*: The user wants to find out which modules can be affected by a change in the current module.

2) *Implementation*: The same data is used for the *Show Used Modules* comprehension, but we traverse the graph in the opposite direction, from the imported module to the importing.

V. DESIGN

The HaskellCompass tools is made up of two parts, the analyzer and the service. The analyzer collects information about the Haskell sources using the compiler while the service integrates it into the CodeCompass system and displays the results to the user.

The analysis is initialized by the `CodeCompass_parse` executable. It runs GHC on the source files either by invoking the normal build tooling or by replaying the recorded commands of a previous build. The command line arguments that invoke the analysis plugin are supplied to the compiler.

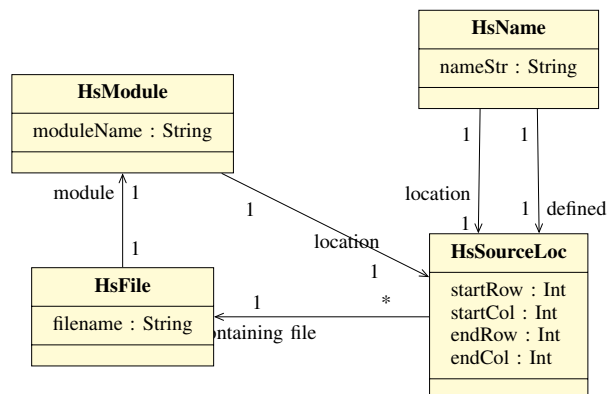


Fig. 4. An excerpt from the UML model of the database schema of HaskellCompass

As it was already described before, the HaskellCompass analyzer sets up hooks in each of the compilation steps using the plugin system. The analyzer then traverses the differently annotated syntax trees.

The analyzer needs to perform certain steps on AST (Abstract Syntax Tree) nodes of certain types. To do that we utilize uniplate, a library for writing simple and concise generic operations [11]. Since uniplate is not suited for heterogeneous traversals, this forces us to make multiple traversals if we are looking for nodes of different types. If this causes a performance problem in the future, we plan to replace uniplate traversals with our own classyplate library [12], that supports heterogeneous traversals.

We stored the data collected during the traversal of the different GHC representations into an SQL database. This was decided in order to follow how CodeCompass works.

The schema of the database extends the basic schema of CodeCompass. While files for example are recognized by the framework regardless of the programming language, tables for most data, like names and types, are re-defined, since they contain different fields than their C-specific counterparts.

Some elements, for example, preprocessor usage are present for Haskell as for C language, they should be reused.

VI. IMPLEMENTATION

HaskellCompass is mostly implemented in Haskell, to be easily integrated with GHC. A few parts of the implementation that are connected with the CodeCompass system are written in C++ instead.

The analyzer can work with SQLite and PostgreSQL databases. The database is connected to the Haskell sources through the persistent [13] package for type-safe, multi-backend data serialization. In Figure 4 you can see an excerpt of the UML diagram that presents the database schema.

The Haskell service is set up as a server using Thrift, which provides Remote Procedure Call (RPC) between the server and the client.

VII. EVALUATION

To test the analysis on a larger-scale project, we ran it on the largest Haskell codebase we could find, GHC's source code. Our goal was to measure the cost of the analysis in addition to the normal build process.

Since GHC is self-compiling, the actual build runs for several stages, each stage building the next compiler that is used to compile the source code again. Since our analysis method requires the usage of already written Haskell code (the analysis plugin), we only used it for stage zero, where the host compiler (a previous version of GHC) compiles the GHC's source code.

At the time of the measurements, GHC repository consists of 384,783 lines of Haskell code according to openhub.net. This codebase also contains tests and other non-built parts of the repository, but most of it is built as part of the build process.

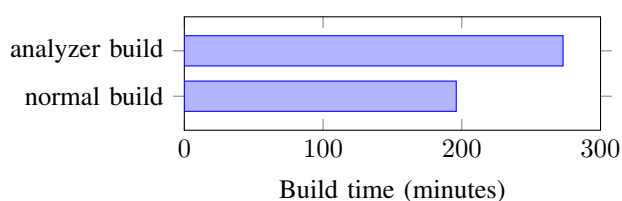


Fig. 5. Runtime performance of analyzer

The result of the measurement (on Figure 5) was 196 minutes for the pure build and 273 minutes for the build with the analysis performed. A 40% overhead may seem large, but for an implementation that is not yet optimized, we think that it is better than expected. The database used for recording the result was 90MBs, and contained more than 1.5 million rows. The measurement was recorded on a virtual machine, using 8GB memory and a 4-core i7 2.4GHz CPU.

This measurement shows that the cost of the analysis is not insignificant, but it compares well with the overall runtime of a build.

On the other hand, the runtime of the actual code comprehension actions is not significant as it uses pre-processed data from the analysis.

VIII. CONCLUSIONS

In this paper we presented a Haskell plugin for the CodeCompass tool that is able to analyse large Haskell projects and provide useful information for the developers. We have suggested useful comprehension queries and diagrams that enhance developer efficiency. We also demonstrated how it can be implemented and provided measurements for its cost in the form of increased compile time.

A partial implementation of the HaskellCompass plugin can be found at github.com/nboldi/codecompass-haskell-plugin

We plan to evaluate the feedback on the initial code comprehension features and to suggest and design new ones. We would also like to test our tool on a wide selection of projects.

REFERENCES

- [1] Zoltán Porkoláb, Tibor Brunner, Dániel Krupp, and Márton Csordás. 2018. Codecompass: an open software comprehension framework for industrial usage. In Proceedings of the 26th Conference on Program Comprehension (ICPC '18). ACM, New York, NY, USA, 361-369. DOI: <https://doi.org/10.1145/3196321.3197546>
- [2] Zoltán Porkoláb, Tibor Brunner. 2018. The CodeCompass Comprehension Framework. In ICPC 18: 26th IEEE/ACM International Conference on Program Comprehension, May 2728, 2018, Gothenburg, Sweden. ACM, New York, NY, USA, Article 4, 4 pages.
- [3] GHC 8.6.5's User Guide, 15.3.5: Source Plugins https://downloads.haskell.org/~ghc/latest/docs/html/users_guide/extending_ghc.html#source-plugins
- [4] Marlow, S., & Jones, S. P. (2004). The glasgow haskell compiler.
- [5] Angelov, K., & Marlow, S. (2005, September). Visual Haskell: a full-featured Haskell development environment. In Proceedings of the 2005 ACM SIGPLAN workshop on Haskell (pp. 5-16). ACM.
- [6] Nicklisch-Franken, J. (2009). Leksah: An Integrated Development Environment for Haskell.
- [7] Chitil, O. (2008). HeatAn Interactive Development Environment for Learning & Teaching Haskell.
- [8] Simon Marlow. 2002. Haddock, a Haskell documentation tool. In Proceedings of the 2002 ACM SIGPLAN workshop on Haskell (Haskell '02). ACM, New York, NY, USA, 78-89.
- [9] Gill, A., & Runciman, C. (2007, September). Haskell program coverage. In Proceedings of the ACM SIGPLAN workshop on Haskell workshop (pp. 1-12). ACM.
- [10] King, D. J., Hall, J., & Trinder, P. (1998, September). A strategic profiler for glasgow parallel Haskell. In Symposium on Implementation and Application of Functional Languages (pp. 88-102). Springer, Berlin, Heidelberg.
- [11] Uniplate library <http://hackage.haskell.org/package/uniplate>
- [12] Classyplate library <http://hackage.haskell.org/package/classyplate>
- [13] Persistent package <http://hackage.haskell.org/package/persistent>

Automated configuration of a Linux web server security

Michal Olenčin

*Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
michal@olencin.com*

Ján Perháč

*Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
jan.perhac@tuke.sk*

Abstract—This paper is dedicated to the design and implementation of automated security configuration after installation of a Linux web server. We analyze cybersecurity, focus, and scope of the configuration. Based on that, we design an automated security configuration with a detailed description. The main accomplishment of our work is a configuration script. The script is compared and evaluated with existing solutions and it achieved the best rating in security audits.

Index Terms—Linux, webserver, security, configuration

I. INTRODUCTION

This paper is focused on designing a secure configuration of a web-based Linux after installation. At first, we analyze the current state of cybersecurity. Then we specify a focus of security. After the analysis, we propose a design of our configuration according to a wide range of resources. The output of the implementation is an open-source script [1] of an automated configuration implemented according to the proposed design. At last, we evaluate the implemented configuration. That consists of a functional test, a compatibility test of the webserver configuration and security audits of default and designed configurations. We run a security audit by the Lynis tool and we also measure and compare a score of existing open-source solutions.

One of our long-term efforts is dedicated to increase computer security using intrusion detection systems (IDS) [2], [3], distributed intrusion detection systems [4], hybrid honeypot technology [5], [6]. We have proposed a formal logical model of IDS [7], [8], using linear logic [9]–[11], and category theory [12]–[14]. We plan to develop an interactive web environment for teaching purposes, where we will implement the results of our research.

II. MOTIVATION

The cybersecurity and protection of sensitive data is a frequently discussed topic today. There is a growing interest in this area mainly due to the increase in cyberattacks and theft of sensitive data.

After installation of a web server, the basic configuration is used and it is necessary to configure it as needed. The basic server setup includes many security issues that need to be corrected by an appropriate configuration for security reasons.

The basic configurations are customized to maximize support and make development easier. The configuration process can be automated using programs written in a scripting language. Creating an automated configuration makes the process of configuration quicker and more effective.

Currently, there are very few solutions dealing with the automation of a Linux web server configuration in the desired complexity and adequate protection. The main goal of this work is to create a solution on an appropriate level of protection and complexity by creating an automated configuration of a Linux web server.

III. THE CURRENT STATE OF CYBERSECURITY

We have performed the current state of cybersecurity according to following reports

- “Cisco 2018 Annual Cybersecurity Report” by Cisco [15],
- “Executive Summary - 2018 Internet Security Threat Report” by Symantec [16],
- “Cloud Security Report” by Alert Logic [17].

The most interesting results e.g. occurrences of malicious programs dedicated to various areas increased as following:

- cryptocurrencies by 8500%,
- IOT devices by 600%,
- ransomware attacks by 46%,
- malware on mobile devices by 54%.

A. Used technologies

In the first step, we have compared the usage of current technologies. We have taken into account the following criteria:

- a Linux distribution,
- a web service,
- database service,
- programming language.

Our main goal was to create a configuration for the wide group of potential users, therefore our main criterion is the popularity of technology.

Because of long-term support and the high market share, we have chosen an Ubuntu Server as the supported distribution.

The version Ubuntu Server 16.04.5 64-bit have been chosen, because of its stability. Ubuntu has 37.8% market share altogether. If the Debian distribution would be added as a supported distribution, the solution would have covered 61.1% of the whole market.

We have chosen an Apache webserver a supported web server, because of its 45.5% market share in its latest version Apache 2.4. The choice of the Nginx as supported web server was also taken into account, but based on the balance between quantity with quality, Apache won. Both are comparable in performance, flexibility and reliability [18]. If the Nginx web server would be added as a supported web server, the solution would have covered $\frac{3}{4}$ of the market share.

As the supported database service, MariaDB was chosen in its latest version of a database server, specifically MariaDB 10.3. The selection of MySQL and PostgreSQL as the supported database server was also taken into account. PostgreSQL is robust, has a complex configuration and is less powerful compared to others. MariaDB is a fork of MySQL, which is 5% more powerful [19] than MySQL. Also, its popularity tends to increase. But both, MySQL and PostgreSQL, are efficient and reliable database services and their support could be added in the future.

PHP was chosen as the supported programming language in the latest version, because of its dominance on the market with 78.9%.

Results are summarized in Table I “*Result of selected security focus*” covering a wide range of usability.

TABLE I
RESULT OF SELECTED SECURITY FOCUS

Scope	Focus
Linux distribution	Ubuntu Server 16.04.5 64-bit
Web service	Apache 2.4
Database service	MariaDB 10.3
Programming language	PHP 7.2

B. Existing solutions

There are a several of automated Web Linux server configuration solutions. Unfortunately, most solutions are lightly complex and only address the configuration of the Apache web service itself, or address a small configuration range. The solutions with the required complexity currently exist in a small amount.

Among the proprietary license solutions, the Lynis Enterprise solution is available [20]. Unfortunately, this solution is too complex for common needs and requires the purchase of a license, which is his disadvantage. For instance, solution offering security audit, system security, vulnerability management, ample support for Unix operating systems, macOS and many more.

Among the open-source solutions, the “*JShielder*” [21] and the “*Ubuntu hardening*” [22] solutions are available. Both solutions are not overwriting existing configurations and do not

use the latest versions of services. Ubuntu hardening solution also includes a smaller range of configuration compared to the JShielder solution.

Each of mentioned solutions is implemented as bash script.

While analyzing existing solutions, deficiencies have been identified which proposed solution seeks to eliminate. The proposed solution compared to existing solutions includes:

- configuration backup,
- use of the latest versions of services,
- it overwrites existing configurations,

with open-source availability, with the required complexity and configuration range for basic needs.

IV. DESIGN OF OUR SOLUTION

The design of our solution is proposed according to the CIS Ubuntu Linux 16.04 LTS Benchmark [23], the CIS Distribution Independent Linux Benchmark [24], the CIS Apache HTTP Server 2.4 Benchmark [25], the book “*Practical Apache, PHP-FPM & Nginx Reverse Proxy: How to Build a Secure, Fast and Powerful Webserver from scratch*” [26], the configuration instructions from the non-profit organization Mozilla, the proposed Lynis configuration, the proposed Tiger configuration and the recommended set of rules for security2 module from the non-profit organization OWASP.

The design is targeted to achieve the highest possible security for common web servers and includes:

- Installation and hardening of the Apache web server and also security2, SSL, evasive, headers, include, http2 and reqtimeout modules.
- Installation and hardening of the MariaDB database server.
- Installation and hardening of the PHP programming language.
- Installation and hardening of the OpenSSH server.
- Security configuration of an operation system:
 - Hardening of directory permissions.
 - Hardening of the file system.
 - Hardening of network protocols.
 - Hardening of the kernel.
 - Hardening of passwords.
 - Hardening of the firewall.
 - Hardening of other areas.

The design of configuration for the Apache webserver includes adding sources with the latest stable version of a package manager, turning off unnecessary modules, limiting leaked information through web server response, set up the service to run under a separate user, configuring security modules, request limits, logging level, permission rights for the service files, etc. Table II presents our configuration of the MariaDB in detail.

Next, the configuration of the MariaDB database server includes adding a source with the latest stable version of a package manager, running the built-in secure installation, preventing access to services outside the local network and access to local files through the database and configuring the

TABLE II
CONFIGURATION OF THE APACHE

Parameter	Value
DocumentRoot	/var/www/html/
FileETag	None
Timeout	10
LimitRequestBody	102400
LimitRequestFieldSize	1024
LimitRequestline	512
LogLevel	notice core:info
Protocols	h2 http/1.1
ServerTokens	Prod
TraceEnable	off
SSLCipherSuite	<i>Based on SSL generator</i>
SSLCompression	off
SSLHonorCipherOrder	on
SSLInsecureRenegotiation	off
SSLProtocol	all -SSLv3 -TLSv1 -TLSv1.1
SSLSessionTickets	off
SSLStaplingCache	"shmcb:/var/run/ocsp(128000)"
SSLStaplingResponderTimeout	5
SSLStaplingReturnResponderErrors	off
SSLUseStapling	on

port. Table III presents our configuration of the MariaDB in detail.

TABLE III
CONFIGURATION OF THE MARIADB

Parameter	Value
bind-address	127.0.0.1
general-log	0
local-infile	0
port	<i>Depends on input</i>

Furthermore, the configuration of the PHP programming language includes also adding a source with the latest stable version of a package manager, disabling a debug and error messages, also disabling dangerous functions, limiting leaked information through web service, configuring session, cookies, the limit of access to system files, etc. Among dangerous functions belongs for example process control functions, functions of the POSIX standard, program execution functions, functions that provide sensitive information and much more. Table IV presents our configuration of the PHP in detail.

TABLE IV
CONFIGURATION OF THE PHP

Parameter	Value
allow_url_fopen	0
allow_url_include	0
disable_functions	<i>Depends on design of configuration</i>
display_errors	0
display_startup_errors	0
expose_php	0
html_errors	0
log_errors	1
open_basedir	/var/www/html
post_max_size	100K
session.use_strict_mode	1
session.cookie_httponly	1
session.cookie_secure	1

Configuration of the OpenSSH server includes installation of the package with a model for server, limiting leaked information through services, port configuration, configuring user authorization, logging level, authorization message, cryptography algorithms and much more. The version 7.2 is considered. Table V presents our configuration of the OpenSSH in detail.

TABLE V
CONFIGURATION OF THE OPENSSSH

Parameter	Value
ChallengeResponseAuthentication	no
GSSAPIAuthentication	no
HostbasedAuthentication	no
PasswordAuthentication	no
PubkeyAuthentication	yes
RhostsRSAAuthentication	no
RSAAuthentication	no
Banner	/etc/issue.net
AllowAgentForwarding	no
AllowGroups	ssh
AllowTcpForwarding	no
Ciphers	<i>Depends on design</i>
ClientAliveInterval	300
Compression	no
DebianBanner	no
IgnoreRhosts	yes
KexAlgorithms	<i>Depends on design</i>
LoginGraceTime	60
LogLevel	VERBOSE
MACs	<i>Depends on design</i>
MaxAuthTries	2
MaxSessions	2
PermitRootLogin	no
Port	<i>Depends on input</i>
Protocol	2
TCPKeepAlive	no
UseDNS	no
UsePrivilegeSeparation	SANDBOX
X11Forwarding	no

Operation system configuration consists of following:

- Hardening of directory permissions includes the configuration of sticky bit on all world-writable directories, set of basic permissions rights and configuring the permission for the sensitive system files.
- The file system is in this design configured so that it includes disabling some unnecessary files systems, configuring USB devices and mounting partitions.
- Hardening of network protocols includes disabling some unnecessary network protocols.
- Hardening of the kernel consists of enabling address space layout randomization, disabling SysRq key, configuring packets and so on.
- Hardening of other areas includes uninstalling unnecessary packages, installing additional security packages and so forth.
 - The following packages belong to the additional security packages included: acct, aide, aide-common, apt-show-versions, arpwatch, auditd, clamav, clamav-daemon, debsums, fail2ban, htop, ntp, sysstat, unattended-upgrades and usbguard.
 - The design also configures some additional security

packages, e.g. auditd, unattended-upgrades and usb-guard.

- Among unnecessary packages belong the following: avahi-daemon, binutils, cups, cups-common, gcc, git, make, snapd and telnet.

The designed configuration, however, might be incompatible with some web applications and websites. It is caused by individual requirements for the configuration of those applications and websites. Additional configuration adjustments are needed after the automated configuration is applied.

V. IMPLEMENTATION AND EVALUATION OF OUR SOLUTION

Our solution is implemented in the form of a bash script, and it is implemented according to the designed configurations. The implementation also includes a password generator. This generator is useful when SSH key is generating and password for MariaDB root account is set up. Further, the whole source code is commented for easy understanding and orientation. In the implementation, the port for OpenSSH and MariaDB is obtained from the user. If the user presses the enter key, the default port is set.

Furthermore, the evaluation was done on a virtual private server for the most realistic conditions and also on a virtual machine. As per functional testing, the script is fully functional on Ubuntu Server 16.04.5 distribution, but also in its newer version 18.04.2. Later we have tested our solution on the newest version for the latest non-lts version Ubuntu 19.04. Our solution for this version is also fully functional. We have selected the Oracle VM VirtualBox as a hosted hypervisor for a virtual machine. For a cloud infrastructure, the DigitalOcean provider was selected.

After the compatibility test of the web server configuration, we have found out that the designed configuration requires small additional configuration adjustments depending on a web application or a website. The compatibility test was done on both types of web pages, with static and also dynamic content.

Security audits were performed using Lynis tool, Nmap tool, SSL Server Test by the Qualys, Mozilla HTTP Observatory tool and Mozilla SSH Observatory tool. Results of security audits are summarized in Table VI “*Result of all security audits*”.

TABLE VI
RESULT OF ALL SECURITY AUDITS

Security audit	Designed configuration	Default configuration
Lynis	97 / 100	56 / 100
Nmap	a few	a lot
SSL Server Test	A+	B
Mozilla HTTP	A+	F
Mozilla SSH	A	C

In the security audit by the Lynis tool was also measured a score of existing open-source solutions. These measurements have been adapted to achieve as highest score as possible. The Lynis security audit was performed with the version 2.7.1 of

the tool. We have compared our solution with other existing configurations, and default configuration with the following results:

- The default configuration has achieved a score of **56 / 100**.
- The Ubuntu hardening solution has achieved a score of **72 / 100**.
- The JShielder solution has achieved a score of **89 / 100**.
- Our solution of configuration has achieved a score of **97 / 100**, which is the best rating when compared to existing solutions.

Unfortunately, the designed configuration did not achieve full score because of the absence of GRUB password for booting, presence of an antivirus program and separate mounting of partitions /tmp, /home and /var. Creating a GRUB password is unwanted on remote computers. Configuring separate mounting of partitions is performing during the installation of the operating system. Additional configuration of mounting partitions is complicated. In the designed configuration is installed a ClamAV as antivirus, but the security audit by the Lynis tool did not detect it.

The security audit for the Nmap tool relieved much less sensitive information leaked for the designed configuration compared to the default configuration, such as information of server distribution and a version of Apache.

The security audit performed by the SSL Server Test scored the designed configuration with an **A+** rating. The default configuration achieved a **B** rating for using weak cryptography algorithms and not providing forward secrecy to all browsers. With the security audit by the Mozilla HTTP Observatory tool the designed configuration has achieved an **A+** rating. The default configuration achieved an **F** rating for missing security-related web service response headers. In the security audit by the Mozilla SSH Observatory tool, the designed configuration has achieved an **A** rating. The default configuration achieved a **C** rating for using weak cryptography algorithms.

Results of security audits are summarized in Table VII “*Result of Lynis security audit*”.

TABLE VII
RESULT OF LYNIS SECURITY AUDIT

Solution	Score
Implemented solution	97
JShielder	89
Ubuntu hardening	72
Default configuration	56

To summarize the implemented solution has the following advantages compared to other existing solutions:

- The design is drawn up according to the use of a large number of resources. For instance, CIS benchmarks guidelines, the proposed Lynis configuration, etc.
- The implemented solution is designed with the required complexity and configuration range for common needs.
- The implemented solution is free and open-source.

- The design uses the latest versions of services.
- The design overrides existing configurations.
- The design includes configuration backup.

VI. CONCLUSION

In conclusion, the result of this work is a fully functional configuration script (available at [1]), which in comparison to existing solutions achieved the best rating in Lynis security audits.

The script is easily expandable and editable, covers a large scope of applications and utilities by market share, comprehensively solves the security of the webserver and the operating system as a whole and overrides existing configurations.

On another hand, the script requires additional configuration, requires a knowledge of Linux configuration and partially rewriting the configurations of the Apache web service.

The work may be extended by including the compatibility for Debian distribution, including the Nginx web service support and including support for other database services such as MySQL and PostgreSQL.

However, the security analysis tools used do not cover the full range of potential vulnerabilities. The security of an entity depends on the weakest element. The designed solution has improved Web server security against known vulnerabilities, but vulnerability to unknown errors cannot be eliminated. In addressing the security problem, care must be taken to continually improve and mend the security of existing solutions, monitor these solutions and maintain cyber hygiene.

ACKNOWLEDGMENT

This work was supported by the following projects:

- Faculty of Electrical Engineering and Informatics, Technical University of Košice under the contract No. FEI-2018-59: Semantic Machine of Source-Oriented Transparent Intensional Logic.
- Slovak Research and Development Agency under the contract No. SK-AT-2017-0012: Semantics technologies for computer science education.

REFERENCES

- [1] M. Olenčin, "Alfavio/Automated-securing-of-Linux-web-server v.1.0.0," June 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.3240990>
- [2] J. Perháč and D. Mihályi, "Intrusion detection system behavior as resource-oriented formula," *Acta Electrotechnica et Informatica*, vol. 15, no. 3, pp. 9–13, 2015.
- [3] D. Mihályi and V. Novitzká, "A coalgebra as an intrusion detection system," *Acta Polytechnica Hungarica*, vol. 7, no. 2, pp. 71–79, 2010.
- [4] L. Vokorokos, A. Baláž, and M. Chovanec, "Distributed detection system of security intrusions based on partially ordered events and patterns," in *Towards Intelligent Engineering and Information Technology*. Springer, 2009, pp. 389–403.
- [5] E. Chovancová, N. Adám, A. Baláž, E. Pietriková, P. Fecil'ak, S. Šimoňák, and M. Chovanec, "Securing distributed computer systems using an advanced sophisticated hybrid honeypot technology," *Computing and Informatics*, vol. 36, no. 1, pp. 113–139, 2017.
- [6] M. Chovanec, E. Chovancová, and M. Dufala, "Dids based on hybrid detection," in *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2014, pp. 79–83.
- [7] J. Perháč, D. Mihályi, and L. Mat'áš, "Resource oriented bdi architecture for ids," in *2017 IEEE 14th International Scientific Conference on Informatics*. IEEE, 2017, pp. 293–298.
- [8] J. Perháč, D. Mihályi, and L. Mat'áš, "Elimination of network intrusions via a resource oriented bdi architecture," *Open Computer Science*, vol. 8, no. 1, pp. 173–181, 2018.
- [9] J. Perháč, D. Mihályi, and V. Novitzká, "Between syntax and semantics of resource oriented logic for ids behavior description," *Journal of Applied Mathematics and Computational Mechanics*, vol. 15, no. 2, pp. 105–118, 2016. [Online]. Available: <http://dx.doi.org/10.17512/jamcm.2016.2.12>
- [10] E. Demeterová, D. Mihályi, and V. Novitzká, "A categorical model of predicate linear logic," *Journal of Applied Mathematics and Computational Mechanics*, vol. 14, no. 1, pp. 27–42, 2015.
- [11] L. Vokorokos, Z. Bilanová, and D. Mihályi, "Linear logic operators in transparent intensional logic," in *2017 IEEE 14th International Scientific Conference on Informatics*. IEEE, 2017, pp. 420–424.
- [12] W. Steingartner and V. Novitzká, "Coalgebras for modelling observable behaviour of programs," *Journal of applied mathematics and computational mechanics*, vol. 16, no. 2, pp. 145–157, 2017.
- [13] W. Steingartner, D. Radakovic, F. Valkosak, and P. Macko, "Some properties of coalgebras and their rôle in computer science," *Journal of Applied Mathematics and Computational Mechanics*, vol. 15, no. 4, pp. 145–156, 2016.
- [14] W. Steingartner, V. Novitzká, M. Bačíková, and Š. Korečko, "New approach to categorical semantics for procedural languages," *Computing & Informatics*, vol. 36, no. 6, 2017.
- [15] Cisco, "Cisco 2018 annual cybersecurity report," February 2018. [Online]. Available: <http://www.cisco.com/c/en/us/products/security/security-reports.html>
- [16] Symantec, "Executive summary - 2018 internet security threat report," March 2018. [Online]. Available: <http://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf>
- [17] A. Logic, "Cloud security report," 2017. [Online]. Available: <http://www.alertlogic.com/assets/industry-reports/alertlogic-cloud-security-report-2017.pdf>
- [18] D. Kunda, S. Chihana, and M. Sinyinda, "Web server performance of apache and nginx: A systematic literature review," *IISTE*, vol. 8, no. 2, March 2017. [Online]. Available: <http://www.iiste.org/Journals/index.php/CEIS/article/view/35842/36837>
- [19] R. J. Dyer, *Learning MySQL and MariaDB*, 1st ed. O'Reilly Media, April 2015.
- [20] CISOfy, "Lynis enterprise," 2019. [Online]. Available: <https://cisofy.com/lynis-enterprise/>
- [21] J. Soto, "Jshilder," March 2019. [Online]. Available: <https://github.com/Jsitech/JShilder>
- [22] J. Radon, "Ubuntu hardening," May 2019. [Online]. Available: <https://cloudsecuritylife.com/cis-ubuntu-script-to-automate-server-hardening/>
- [23] CIS, "Cis ubuntu linux 16.04 lts benchmark," December 2017.
- [24] CIS, "Cis distribution independent linux benchmark," December 2017.
- [25] CIS, "Cis apache http server 2.4 benchmark," July 2018.
- [26] A. Ling, *Practical Apache, PHP-FPM & Nginx Reverse Proxy: How to Build a Secure, Fast and Powerful Webserver from scratch*, 1st ed. Adrain Ling Kong Heng, May 2015.

Managing The Implementation Of A Smart Pedestrian Network (SPN) System

George Papageorgiou
E.U.C. Research Center
Nicosia, Cyprus
g.papageorgiou@euc.ac.cy

Anastasia Ioannou
E.U.C. Research Center
Nicosia, Cyprus
ai185155@students.euc.ac.cy

Athanasios Maimaris
E.U.C. Research Center
Nicosia, Cyprus
a.maimaris@cycollege.ac.cy

Alexander N. Ness
University of Oslo
Oslo, Norway
alexnn@student.sv.uio.no

Abstract—This paper presents a Smart Pedestrian Network (SPN) System that is geared towards promoting sustainable mobility. The SPN system has been designed to reach multiple market segments by providing information on suitable walking routes aiming to satisfy potential users' needs. The paper presents the main objectives of the SPN system as well as its system structure, application features, functions, and data sources. By doing so, we showcase how the SPN system has been developed. Furthermore, a framework is proposed so that the SPN system is successfully implemented in a variety of urban conditions. The proposed framework emphasizes the element of adaptability, so that SPN can be adjusted where and when necessary, to deal with a variety of contexts and specific sustainable mobility issues, depending on the particular municipality conditions.

I. INTRODUCTION

With an estimated 4.2 million deaths each year been directly attributed to air pollution [1], it is evident that our health is at great risk. Air pollution mainly stems from high levels of vehicle use. Current ecosystems and ocean life are also being adversely affected by CO₂ emissions and the resulting climate change. If we do not find a way to deal with this problematic situation, life as we know it will eventually cease to exist.

Further, our attitudes and actions today, will have a direct effect on the lives of future generations. Something that can be done, is to develop more sustainable means of transport to reduce pollution and global warming. In this way, we can significantly mitigate the climate change crisis by maintaining cleaner air in our cities, preserving our natural environment and land, and therefore, conserve our planet for future generations.

In a world of global warming and unsustainable growth, new approaches to transportation and mobility must be considered. An effective way to reduce pollution and global warming is to develop sustainable urban policies to substitute vehicle use with active means of transport. Promoting walking where and when possible is a plausible solution to our problem. On the other hand, this is not an easy solution to implement as people still opt to take the presumed more convenient and easier route, which is using a motorized vehicle instead of engaging in active mobility.

This can be attributed to lack of awareness of appropriate walking routes as well as the lack of knowledge of the many benefits that active mobility can have both to ourselves, and to our planet. In the current transport model used in our cities,

we see that we heavily rely on vehicle use [2]. There is a lot to be gained if we embark on more active means of transport such as simply walking. With walking, we can have a plethora of benefits for ourselves and the environment. In fact, the ancient Greek physician Hippocrates, rightly argued that walking is man's best medicine! The key to this sustainable mobility transformation, is to encourage, convince but also support people when engaging in active mobility.

Current advancements in Information and Communication Technology (ICT) and Artificial Intelligence (AI), create a big opportunity to really support active mobility [3]. For example, a smartphone application system could be developed so that people are informed about appropriate walking routes, but also give feedback via crowd-sourcing, to municipal authorities on issues regarding the pedestrian network conditions. Even though several applications may give general information on the pedestrian network environment, they are not necessarily geared towards suggesting appropriate and enjoyable walking routes, and thereby encouraging people to walk more. Also, they do not provide information on safer routes, and are unable to inform users about amenities such as water fountains, public toilets or resting places that certain user groups such as elderly people may particularly need.

We posit a dedicated pedestrian smartphone application to overcome the limitations posed by existing software applications. Our purpose is to satisfy the needs of many potential pedestrian user groups, by providing them with relevant information about the pedestrian network. Moreover, through increased awareness, we would promote walking, and the substitution of vehicle use with active means of transport. It is important to make known to potential users the benefits they can reap from using the proposed Smart Pedestrian Network (SPN) app. Such benefits are exhibited in fig. 1. Information about these benefits is incorporated in the proposed SPN application. In this way, we continuously motivate people to use SPN and thereby, walk more.

The proposed SPN application is able to personalize the user interface in order to satisfy specific user needs. For instance, elderly people may wish to be shown a route with amenities such as public toilets and rest places. A tourist may wish to take a longer route that has better sightseeing or cultural events taking place. The SPN app, via Artificial Intelligence (AI) algorithms, is able to provide such tailored route suggestions.

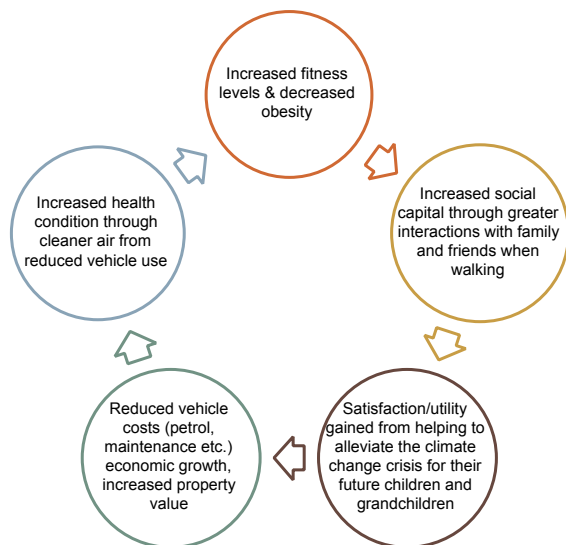


Fig. 1. Benefits from Implementing SPN

Current smartphone navigation systems have not been devised to aid such user groups with such specific needs [3]. This is where the SPN app comes in. It has been designed to satisfy these unfulfilled market needs. Put simply, we devise an innovative software system that provides the necessary information and tools to promote walking. In this way, we are enabling a feeling of safety and comfort within the urban pedestrian environment.

Even though the SPN is an innovative system, that on its own does not guarantee a successful implementation in cities around Europe. It is important therefore to consider the use of an appropriate framework for properly implementing SPN. As shown later in the literature review, current methods for software development which include the Waterfall Model and the Agile Model, do not adequately address these implementation issues. Hence, in this paper, we investigate such issues and suggest a road map for successfully implementing our SPN system.

Our paper is structured as follows; in the next section we examine past literature and relevant studies on software development methods and the potential for mobility navigation systems. We also discuss here the effectiveness of methods to support the successful development of software systems. In section III, we outline the SPN system structure and present the proposed implementation framework in section IV. Finally, in section V, we summarize our main conclusions and recommendations regarding our experience from developing the SPN system.

II. LITERATURE REVIEW

According to research studies, as of 2018 [4], the majority of smartphone owners use a navigation application to help them in tracking a suitable route for their journey. The Manifest study [5] surveyed more than 500 smartphone owners to

understand how people are relying on technology more to get from point A to B. Their findings were as follows:

- Over three-fourths (77%) of smartphone owners regularly use navigation apps.
- Google Maps (67%) is the most popular navigation app by a wide margin. It is the preferred navigation app for nearly 6x more people compared to the second-most popular app Waze (12%).
- One-quarter (25%) of smartphone owners say they use their preferred navigation app because they believe it offers better directions.
- Over one-third (36%) of smartphone owners use navigation apps prior to leaving their location.
- 34% use them en route, and 30% use them both prior to leaving and en route equally.
- An overwhelming number of respondents (87%) use navigation apps for driving directions the most
- People use navigation apps for non-driving directions such as public transportation, walking, or cycling twice as much in the Northeast than in other U.S. regions. This might be due to cultural reasons and better infrastructure conditions.

From the above, we can derive several implications for the development of navigation systems suitable for pedestrian use. The fact that an overwhelming 87% of respondents use navigation applications primarily for driving directions, is a disturbing fact. This fully supports our argument that there is an extremely high level of vehicle use in our cities. Furthermore, it provides evidence that vehicle use is proving to be a major contributing factor to environmental detriment and the deterioration of human health.

As a result, there is a lot of potential for people to use navigation systems for active transportation such as walking and cycling. This showcases that indeed not only is there demand for such products, but that this demand could be enhanced further if an appropriate smartphone app is developed that appeals to the public. Hence, there is a potential market granting our SPN app, the opportunity to achieve its goal in promoting higher walking levels and reducing air pollution.

Such findings may suggest that the reason so many individuals use navigation applications for vehicle use, is because there currently does not exist in the market an appropriate application designed specifically to promote sustainable mobility. Existing applications fail to completely satisfy particular pedestrian user needs such as getting information on amenities, safe routes, comfortable mobility etc, when engaging in active transportation.

Further, in a market research and concept study for SPN [6], it was shown that there is demand for such a smartphone pedestrian network app so that people are informed about the various aspects of the pedestrian network including comfortable mobility, safety, convenience, and connectivity. Furthermore, it was found that successfully implementing SPN would be challenging as high levels of investments and commitment from the side of the municipal authorities are

required. Therefore, new business models that involve the element of sponsorship and co-creation should be considered.

Therefore, what is important to also stress in this section, is that for the SPN system to be a success, it must be properly implemented using an appropriate framework. This is discussed in the following paragraphs by reviewing the most widely used methodologies for developing information systems. These include the waterfall approach and the agile method.

According to McConnell [7], the waterfall methodology model, is a sequential software development process where progress flows steadily toward the conclusion - like a waterfall - through the phases of a project. That is, analysis, design, development and testing. The nature of the Waterfall methodology insists that each phase be complete and perfected before the start of the next phase. This involves fully documenting a project in advance, including the user interface, user requirements, and all the features' variations and outcomes.

A detailed investigation and full research into a product feature are conducted up front, eliminating (most) project risks. With the bulk of the research done in advance, estimates of the time required for each stage are more accurate, thus providing a more predictable release date. The waterfall method must follow the following steps:

- Analysis
- Design
- Implementation
- Testing

During the analysis stage, the product development team analyzes the requirements, and fully understands the problems. This is a research phase that includes no building. The team attempts to ask all the questions and secure all the answers they need to build the product requirement. In the design stage, the software developers create a technical solution to the problems set out by the product requirements, including scenarios, layouts and data models. This phase is usually accompanied by documentation for each requirement, which enables other members of the team to review it for validation. Once the design is approved, technical implementation begins. This is often the shortest phase because research and design have been done in advance. Upon completion of full implementation, testing needs to occur before the product can be released to customers. The software testing team will use the design documents, personal and user case scenarios delivered by the product manager in order to create their test cases.

Very early on, Royce [8] in a study about managing development of large software systems is criticizing the sequential nature of the waterfall method. It was asserted that software should not be developed like an automobile on an assembly line, in which each piece is added in sequential phases. In such sequential phases, every phase of the project must be completed before the next phase can begin. There are criticisms against this approach in which developers first gather all of a project's requirements, then complete all of its architecture and design, then write all of the code, and so on. Specifically, this approach has been objected to due to the

lack of communication between the specialized groups that complete each phase of work.

This is where the agile method comes in to fill in the gap, as an improved method of software development. According to McConnell [7], the agile methodology is an iterative approach to software development that is performed in a collaborative environment by self-organizing teams which produce the final product version in a sequence of multiple product releases. The methodology produces high-quality software in a cost-effective and timely manner to meet stakeholders' changing needs.

Every product release begins with a list called a "backlog", which consists of a list of prioritized requirements which are dynamic and can change at any time. Determining the next product release, a smaller number of requirements can be chosen from the backlog list. The work involved in the next product release is called "iteration (or sprint)". Each iteration has a short time frame which usually lasts from one to four weeks, depending on the team's experience. A key element of an iteration, is that the priorities regarding which requirements should be built, do not change within the iteration; this list should only change from one iteration to another.

The Agile methodology accepts that project change is inevitable. The use of small iterations allows changes to be absorbed quickly without inflicting significant project risk. The system to be developed can be adjusted as business priorities evolve. With the next iteration, the software development team can adapt to the new priorities.

It is easy to see how the "waterfall" methodology is far from optimizing the final software solution compared to the agile methodology. First of all, it assumes that every requirement of the project can be identified before any design or coding occurs. But in truth, this is seldom the case. This is an impossible task that makes the waterfall model inadequate for effectively developing software solutions. On the other hand, the waterfall model provides structure to software development which is essential for complex projects such as SPN.

Therefore, a combination of the waterfall and agile methodologies would be ideal for the case of developing the SPN information system and smartphone app. We have attempted to follow the waterfall methodology in outlining a roadmap in order to analyse the SPN structure, and future goals we hope to achieve within a given period of time. This was also done for the SPN design, implementation and testing phases. In order to, add flexibility and effectiveness in developing SPN we incorporated the dynamic approach as suggested by the agile methodology model. Therefore, an integrated approach that combines the waterfall and agile models was used as shown in the next section.

Further, we see that current methods for developing software interactive systems, such as SPN fail to effectively consider implementation issues. As we see further on in this paper, we propose an approach to deal with such matters which involves making the system ready for the market via testing, adaptation, feedback and adjustments, to improve the chances of successful implementation.

III. DEVELOPING THE SMART PEDESTRIAN NETWORK (SPN) SYSTEM

We are currently in the process of developing the SPN system which provides real time analysis of the actual physical and virtual environment that exists in a specific area within the pedestrian network. Furthermore, the SPN system assists pedestrians in formulating their interests, preferences or goals at any given moment in time regarding walking routes, specific attributes, objects of interest and their properties. SPN is able to calculate alternative routes based on pedestrian preferences, topography, and actual status of the objects in the physical and virtual environment in real time and present a holistic vision of the route options to pedestrians on-demand. Additionally, it traces walking routes, measure relevant statistics, collect updates regarding actual status and properties of route objects, and present it to pedestrians.

SPN will offer re-routing upon calculation of (unfavorable) changes in the status of particular objects along the route and deviations from the initial route plan and schedule by time, distance, physical condition, health, degree of fatigue, etc. It has also been devised to support final assessment of route results by the pedestrian. This is achieved by collecting important data for statistics, analytics, prediction and feedback. Finally, SPN will process the results and update statistics regarding the particular city condition, and pedestrian experience based on personal profiles.

A. SPN System Structures & Data Sources

The SPN system structure comprises of the pedestrian profile, preferences, filters on key journey information, rules for route suggestions, data sources and specific smartphone app functions.

There are static data and dynamic data that we can utilize. Static data comprise of attributes such as distance, land metrics, traffic infrastructure and so on. Dynamic data on the other hand, comprises of things that are changing over time such as on-going events, maintenance work and changes in weather conditions. Most static data are currently available from open sources such as MAPS.ME [9] and OpenStreetMap [10]. Both static and dynamic data sources could also be given from the local city mapping services.

B. SPN System Features

The SPN system has some important features which are discussed as follows. For example, the SPN app takes into consideration weather conditions, available amenities in the area and safety conditions during the night. In addition, the SPN app is able to calculate the number of steps and provide rewards when the user achieves specific milestones. There is to be an audio option for those who are visually impaired. A sharing journey option creates a sense of community which is available through the smartphone app. In this direction, a chatting service could also be available in order to promote walking together with friends.

Further, in collaboration with local cafes, restaurants and retail stores, rewards would be given to those who walk

more. Also, information about the effect of walking on CO2 reduction is to be provided. Moreover, the SPN system is able to handle the uploading of information by the end-users, thus implementing crowd-sourcing schemes. For the promotion of a sense of community, it could easily be linked to Facebook or other social media platforms. We outline in further detail the main features the SPN system will incorporate:

- **Time of Day, Weather & Location Detection:** The SPN accounts for weather conditions and time of day when considering appropriate walking routes. It also detects the user's location.
- **Amenities:** The SPN is able to suggest appropriate walking routes depending on a list of amenities users wish to have along their journey.
- **Safety:** The application provides the safest routes especially for those travelling alone or during the night.
- **Number of Steps:** Users are able to view the number of steps they have made on a journey per day or week. They could also track their progress over time. The number of steps could also be combined with a free drink in a local café, or other rewards.
- **Audio Guidance:** Users would have the option for audio guidance, particularly for those who may be visually impaired.
- **Save Option:** For frequent journeys taken, users would be able to save their journey and store it. They would also be able to share these saved journeys with others.
- **Group Walks & Chats:** This option would enable users to form groups with friends and families to walk together. A chat would also be available for people to interact with each other. They can use this to arrange meeting points etc.
- **Estimating CO2 Levels:** It should be able to estimate levels of CO2 reduction from walking.
- **Point Collection:** To give incentives to individuals for walking more, the proposed SPN application can potentially collaborate with cafes, stores and other enterprises to reward individuals who walk more. For instance, for every so many steps taken in a month, individuals may be entitled to discounted drinks or food specific products such as discounted pair of shoes from sponsor companies.
- **Crowd-Sourcing:** The SPN system would be able to collect information after users upload data and inform municipalities on problematic areas of the pedestrian network. Users would be able to upload such information via images or location tags.
- **Linking SPN system to Social Media:** For easier use and to share between friends and family SPN can be linked to social media platforms, such as Facebook or WhatsApp.

C. SPN Application & Functions

The user would be able to first state their preferences and make their identity known (i.e. if they are a tourist or local, or perhaps disabled). The SPN application system collects pedestrian preferences using the User Interface functions.

Once the information is submitted by the potential user, the application would respond according to the user's profile.

The system then identifies the user's location. Based on this, the proposed SPN application will list the best routes for users along with media data such as metrics, analytics, pictures, video and recommendations. In this way, the SPN can reveal to the user, the most suitable route options for them. The user then is presented with an 'option' button where they are able to choose the walking route they wish to take. The SPN system enables pedestrians to download and store their local routes as well as trace them. This also allows the users to re-open any saved walking routes whenever they wish even when they are off-line. This of course, would be particularly useful for those who have difficulty accessing their mobile data or have no internet connection.

If any changes are to take place - either along the selected route or from user preferences, then the SPN system would be able to notify the user of these changes and provide alternative routes. In addition, SPN would provide continuous updates to users so that they will be notified of any changes that occur and hence, affects their selected walking route. These updates would be carried out in real time, so that accurate and reliable information is given to users. In such a case, an alternative walking route would also be provided. If the user deviates from their selected route, SPN would inform users about this deviation.

Furthermore, users will be able to evaluate routes and the quality of objects, such as amenities along their walking routes. Such feedback can then be collected by SPN, via the Smart City Monitor (SCM), which collects information from all data sources in real time. This is an important component not only for the users but also for the developers. In this way, we will be made aware of any shortcomings of SPN. By collecting such constructive feedback, any issues can then be rectified. What is more, the data collected can also provide us with great insight into what particular services or features our potential users want to see in the navigation system. Therefore, evaluating such data can also allow us to incorporate these desired features, and thereby, improve SPN.

IV. THE PROPOSED SPN IMPLEMENTATION FRAMEWORK

The proposed framework for implementing SPN is based on an integration of the waterfall model and the agile methodology. It is comprised of two phases. Phase 1 presents aspects of implementing SPN in one particular city, while Phase 2, examines the issues of implementing SPN in multiple cities. In fig. 2 which depicts Phase 1, we show the very beginning of implementing the SPN system. To secure funding and support from municipalities and investors, it is necessary to properly present our business plan. Explanations should be given to our potential stakeholders, why SPN is worth investing in, how it will benefit society and our environment, and how we will get people interested in using SPN. Also, explanations should be given on how the budget is prepared and utilized. Once having attained the necessary support and funding, part of it,

would be allocated for all the necessary tools, equipment and technology that is needed to make SPN successful.

Once an initial version of the SPN system is agreed upon with stakeholders, the next step is to test it. Testing is to be carried out with the cooperation of the municipal authorities in order to adapt SPN to the context and needs of the particular city. Municipal authorities and a group of end users would be required to use the system for a given number of weeks, after which they will provide their feedback. We can use any positive feedback to further promote SPN. For any shortcomings, we will make the necessary adjustments and changes to satisfy particular needs of end users and municipal authorities.

Having had first-hand experience with implementing SPN in Phase 1, we will proceed to Phase 2 as depicted in fig. 3. In Phase 2, the SPN system is ready to be launched in the multiple municipalities that are interested. Priority would be given to municipalities in various European countries. We could begin with the countries of the associated organizations/institutions, which are part of the SPN partnership. These countries are: Austria, Cyprus, Italy and Portugal. In this way, we can better monitor the SPN system's performance.

All necessary data and feedback from stakeholders should be collected. This will be used to assess various factors including:

- how frequently the SPN app is being used on average;
- SPN application performance;
- which markets are using the SPN app more;
- number of weekly/monthly downloads, ratings;
- if the market is increasing via widespread use;

and other factors that deem necessary for monitoring a successful SPN implementation.

This information is to be used in order to notify governments and municipalities about various issues regarding the pedestrian network. Successful collaboration with governments and municipal authorities is essential in order to invest in improving infrastructural conditions, walking paths, providing amenities etc. It is important to consider feedback from stakeholders as this will help in improving the SPN system. Feedback will be taken into consideration and any necessary changes or adjustments will be made to meet and satisfy users' needs.

V. CONCLUSION

This paper presents our experiences from developing a smart pedestrian network (SPN) system for promoting walkability in urban spaces. Even though the system considers effectively the variety of user needs, that does not guarantee its successful implementation. Going through a literature on the main methods of developing information systems, we found that there are gaps when it comes to supporting the implementation of interactive information systems such as the SPN case. As a result, we have developed an implementation framework for SPN that would also be useful for other similar interactive systems. The proposed framework integrates aspects of the waterfall model and the agile software development method.

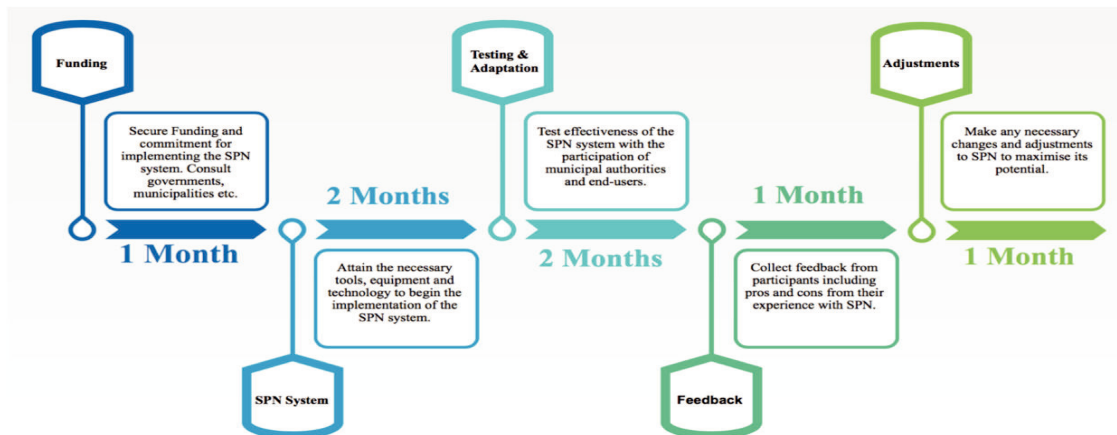


Fig. 2. Roadmap Phase 1

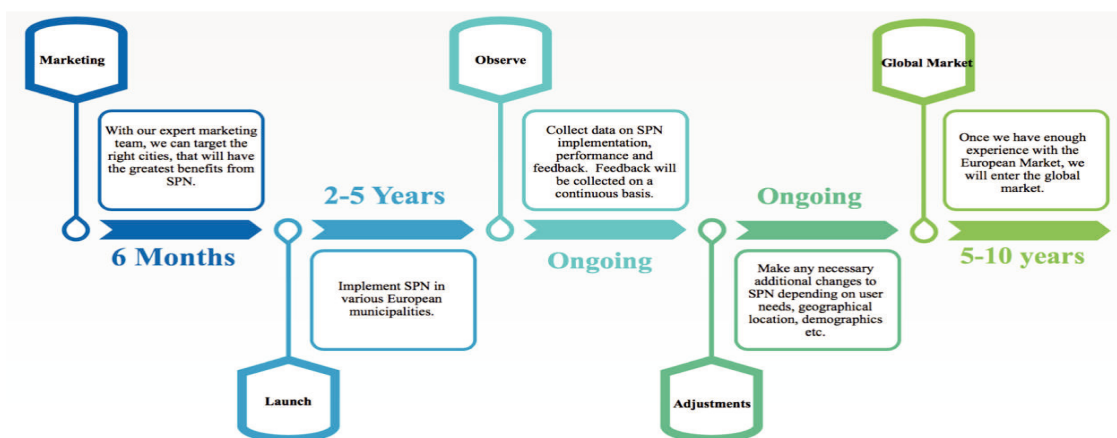


Fig. 3. Roadmap Phase 3

It emphasizes testing, adaptability, feedback and gives room for adjustments to be made based on future conditions. The next step would be to apply the proposed system and evaluate its performance.

ACKNOWLEDGMENT

The research presented in this paper is co-funded by the Republic of Cyprus and the European Regional Development Fund as part of ERANET Cofund Smart Urban Futures (ENSUF) Joint Programming Initiative (JPI) Urban Europe, through the Research Promotion Foundation, protocol no. KOINA/IIKII URBAN EUROPE/1215/11. This framework is supported by the European Commission and funded under the HORIZON 2020 ERANET Co-fund scheme

REFERENCES

[1] World Health Organization. (2019) Air pollution and health. Accessed: 2019-09-01. [Online]. Available: <https://www.who.int/air-pollution>
 [2] G. Papageorgiou and A. Maimaris, "Towards the development of Intelligent Pedestrian Mobility Systems (IPMS)," in *2017 International Conference on Electrical Engineering and Informatics (ICELTICs)*, Oct 2017, pp. 251–256.

[3] E. Conticelli, A. Maimaris, G. Papageorgiou, and S. Tondelli, *Planning and Designing Walkable Cities: A Smart Approach*. Cham: Springer International Publishing, 2018, pp. 251–269. [Online]. Available: https://doi.org/10.1007/978-3-319-77682-8_15
 [4] G. Papageorgiou, A. Maimaris, T. Efstathiadou, and E. Balamou, "EVALUATING ATTITUDES ON THE QUALITY OF SERVICE OF PEDESTRIAN NETWORKS," in *WIT Transactions on The Built Environment*, S. Ricci and C. A. Brebbia, Eds., vol. 176. WIT Press, sep 2017, pp. 35–41.
 [5] R. Panko. The Popularity of Google Maps: Trends in Navigation Apps in 2018. The Manifest. [Online]. Available: <https://themanifest.com/app-development/popularity-google-maps-trends-navigation-apps-2018>
 [6] G. Papageorgiou, D. Demetriou, E. Balamou, and A. Maimaris, "Market research and concept study for a smart pedestrian network application," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, ser. PETRA '18. New York, NY, USA: ACM, 2018, pp. 522–528. [Online]. Available: <http://doi.acm.org/10.1145/3197768.3201550>
 [7] S. McConnel, *Software Estimation: Demystifying the Black Art: The Black Art Demystified (Best Practices)*. Microsoft Press.
 [8] W. W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques," in *Proceedings of the 9th International Conference on Software Engineering*, ser. ICSE '87. Washington, DC, USA: IEEE Computer Society Press, 1987, p. 328–338.
 [9] MAPS.ME. Maps.me. [Online]. Available: <https://maps.me>
 [10] OpenStreetMap. Openstreetmap website. [Online]. Available: <https://www.openstreetmap.org/#map=9/35.1692/33.4355>

A Framework for Evaluating the Benefits of Implementing a Smart Pedestrian Network System

George Papageorgiou
E.U.C. Research Center
Nicosia, Cyprus
g.papageorgiou@euc.ac.cy

Anastasia Ioannou
E.U.C. Research Center
Nicosia, Cyprus
ai185155@students.euc.ac.cy

Athanasios Maimaris
E.U.C. Research Center
Nicosia, Cyprus
a.maimaris@cycollege.ac.cy

Alexander N. Ness
University of Oslo
Oslo, Norway
alexnn@student.sv.uio.no

Abstract—Information and Communication Technology (ICT), can serve as a catalyst for promoting sustainable means of transport. Walking can greatly contribute to reducing pollution and improving health. Furthermore, it fosters social capital and physical activity, thus contributing to a higher quality of life. The implementation of a Smart Pedestrian Network (SPN) information system is proposed. Such an implementation requires the collaboration of various stakeholders including the public, local authorities and local businesses. To convince stakeholders of the viability of implementing SPN, the benefits of walking should be clear. This paper proposes a framework to quantify the active mobility benefits so that stakeholders can assess the investment that can be realized from implementing SPN. The proposed framework makes use of discounted cash flow methods. The benefits are shown to be large enough and very much in favor of investing in technology and implementing an SPN system.

I. INTRODUCTION

Advancements in Information and Communication Technology (ICT), presents a great opportunity for promoting active mobility strategies. It should be noted that ICT is mainly used to optimize vehicle traffic flows via intelligent transportation system (ITS) techniques. This heavy reliance on improving vehicle traffic would have negative effects on active means of transport as it attracts more vehicles in our cities [1]. Specifically, this paper investigates the benefits accrued from the use of ITS to promote active mobility via implementing a Smart Pedestrian Network (SPN) information system in municipalities around Europe.

Active means of transport must be encouraged if we are to address major environmental problems such as pollution and climate change, but also, improve well being and quality of life. With an estimated half a million [2] people in Europe alone dying from diseases triggered by air pollution, it is imperative that local authorities take drastic measures to ensure that pollutants stemming from vehicle use are reduced to zero. Active mobility such as walking can offer multiple benefits to governments, society, the environment and individuals. Currently, air pollution is costing the global economy 4.5 trillion USD annually [3]. A large proportion of this is due to health issues arising from air pollution. By encouraging the substitution of vehicle use with active means of transport, governments and people, can expect to see a significant reduction in healthcare costs. Furthermore, illnesses and premature deaths caused by air pollution impede economic activity and development [4].

Therefore, economic performance also benefits from increased walking levels, if it means that fewer people will be sick and thus absent from work. In other words, active mobility can increase labor productivity.

The contaminated and heavily polluted oceans are also a major issue. Organizations, the media and environmentalists are constantly reminding people and governments of the detrimental effects that human activity is having not only on human life, but on animal life and ecosystems as well [5]. Habitats of the ecosystem are constantly under threat and soon, many species are predicted to be extinct. If we do not immediately attend to the issue of climate change, the planet as we know it, will no longer be sustainable. Promoting walking instead of driving, can greatly help maintain cleaner oceans and ecosystems.

Walking can also help individuals build stronger trust and bonds between family, friends and neighbors [6]. The simple act of walking – an action that is free and readily available to most people – has also shown to improve peoples' social, psychological, emotional and mental well being [7]. This alone exemplifies the myriad of benefits that governments, municipalities, the economy, the environment and individuals can gain from walking.

So far, government intervention has failed to properly address climate change and provide adequate long term solutions. In the past, governments have tried to improve road networks to reduce congestion in the hopes of reducing carbon emissions. However, this solution resulted in unwelcome repercussions; rather than reducing carbon emissions, such policies only increased them due to the fact that more people are now encouraged to drive. If we are to help attenuate the issue of climate change, we must look to more sustainable means of transport.

This is why we posit the implementation of a Smart Pedestrian Network (SPN) information system that will be geared towards encouraging people to walk more, by providing them with suitable walking routes, based on their needs. For instance, elderly people may require special amenities such as benches to rest, or water fountains along their walking routes to keep hydrated. Tourists may wish to walk along a path that entails better sightseeing. SPN will take the form of a smartphone application to provide users with the most appropriate walking routes to satisfy their needs. In addition,

SPN could provide rewards and incentives for people who walk. For instance, by collaborating with cafes, restaurants and other businesses, we can provide special offers to SPN users who walk a specified number of steps.

To evaluate the benefits of walking as well as the potential and profitability of implementing SPN costs, revenues and benefits must be quantified and monetized. It is vital to quantify such factors to assess the magnitude of benefits accrued from walking. In this way, we could motivate stakeholders to support SPN implementation once it is shown that it is an investment worthwhile. In this paper, a framework is presented which serves as a guidance on how to estimate the benefits of walking. These benefits include reduced healthcare costs, reduced fuel costs and reduced monthly expenses. To be able to showcase this, we work on a hypothetical scenario where SPN is implemented in an urban region within Europe. In this way, an illustrative example is provided and evaluated on its application prospects in Europe. The framework makes use of discounted cash flow methods. This is important for investigating the long term impacts the SPN will have on the natural environment, the economy, governments, ecosystems and individuals.

The paper is structured as follows. In section II, the literature review is presented whereby studies evaluating the benefits of walking are discussed. Furthermore, current global issues caused by climate change and vehicle use are also highlighted. In due course, section III discusses the development and implementation of SPN. A specific scenario is set up, whereby SPN aims to target a certain market share within Europe, and how given this, we can calculate all possible benefits. Further, in section III, we also examine all possible monthly costs of running the SPN project, and how SPN will obtain its monthly revenues. Finally, conclusions are given regarding the viability of implementing the Smart Pedestrian Network concept and the many benefits accrued.

II. LITERATURE REVIEW

Health benefits attributed to walking have been studied in great detail. Many studies have found that walking 10,000 steps per day [8] can help individuals suffering from high blood pressure, improve body composition and serum lipid levels. Walking 10,000 steps per day offers benefits similar to other moderate forms of exercise that people may carry out. Therefore, walking can provide a cheaper and easier alternative to other forms of exercise.

Obesity is currently a severe health issue that 50% of men and women in Europe are facing [9]. This not only results in poor quality of life, but possible greater healthcare costs for individuals and governments as well as premature deaths. Obesity has also been linked to increased sedentary behavior and a decline in walking levels. But many studies have found that active mobility like walking can indeed not only help prevent weight gain but also reduce obesity. Senauer and Gemma [10] conducted a study comparing physical activity of Japanese and Americans. A high percentage (67%) of Americans were classified as obese compared to only 25%

of Japanese. When examined why Japan has one of the lowest obesity rates in the world, an important finding was uncovered. Japanese people are more physically active than Americans; but this was not due to engaging in more planned physical activity such as going to the gym or swimming. It was attributed to Japanese people engaging more in active mobility. Senauer and Gemma [10] found that Americans used their vehicles to a much greater degree than Japanese people and suggested that the American government should encourage the public to walk more and use their cars less. Furthermore, Harvard Medical School [11] found that walking briskly for around 1 hour per day reduces the effect of these obesity-promoting genes by 50%. In addition, it was also found that walking can promote other health benefits including reduced cancer risk, joint pain relief and a healthier immune system.

Hill [12], found that walking 30 minutes per day can aid people with diabetes. Currently, 60 million people in Europe have been diagnosed with diabetes [13]. Such discoveries therefore also denote how walking can help diabetics. Hill [12] also cites how using technological equipment such as counting the number of steps via a smartphone to encourage people to walk more can be a great incentive. Implementing SPN could further gamify the experience of walking. Today, counting number of steps is a standard feature of every smartphone. In the study, Hill [12] looked at diabetics who all underwent counselling that aimed to increase their physical activity. Participating patients were given specific targets that was aimed at increasing their walking levels over time. Hill [12] concluded that even those who did not manage to reach their targets saw an improvement in their health, while those who achieved their targets saw a much larger development in their health.

In contrast to what the majority of people may believe about driving to places, vehicle use can actually be an inefficient, time-consuming and costly means of travel [14]. It has actually been found to reduce punctuality of workers. This is all due to the increased probability of a car breaking down, road accidents and traffic congestion. All these factors hinder a persons productivity and efficiency both in their professional capacity and their personal lives. Furthermore, road accidents and cars breaking down also produce negative knock on effects for other individuals who may be travelling along the same route. Contrary to this, active mobility can reduce work absenteeism, increase energy levels and ensure that people reach their destinations in time. Such consequences will thus also positively impact ones career development and progression.

As already mentioned, walking can also contribute to increasing social capital and help build better relationships amongst friends, families and neighbors. Kim and Yang [15] undertook a study to ascertain a possible link between social capital and walking levels. To achieve this, data was obtained from 591 individuals in Seoul, regarding the reasons for walking and participants opinions on the possible gains they got from their walking experience. The most important finding from this study was that strong levels of trust was seen

between neighbors who frequently walked together for leisure compared to those who did not. Hence, Kim and Yang [15] deduced that walking can aid in reinstate social function within neighborhoods.

It is important to grasp how individuals perceive sustainability, and in what way walking can impact social capital. Rogers et al [6] interviewed key informants, municipal decision makers and neighborhood leaders in New Hampshire, Manchester and Portsmouth in the United States. The appropriate demographic data was obtained on neighborhoods in these regions and a social capital survey conceived by [16] was given out to participants. The study is extremely relevant for the purpose of our research as it deals with directly asking participants what motivates them to walk and how high are the trust levels within their communities. The most important factors that motivate people to walk are safety, the provision of sidewalks and lighting. This is also in line with findings from previous research [17] indicating that safety and pedestrian network conditions are crucial determinants of walking. These results are important for policy formulation and implementation. To ensure the right policies are enacted, peoples' mindsets and cognitive behavior must be accounted for. By doing so, we can hope to give incentives to people to walk more by providing them with the right conditions such as safe walking routes and adequate infrastructure.

Ball et al [18] investigated the relationship between aesthetically appealing environments, convenience of walking, the presence of a companion when walking and walking levels. The study did this accounting for differences between men and women and reported mental and physical health. This was carried out by collecting and analyzing cross sectional self-report data from a survey that consisted of 3,392 Australian adults. Further, Ball et al [18] found that men and women who found their environment less aesthetically appealing or less convenient had a lower probability of walking for exercise or recreation compared to those who found their environment more aesthetically pleasing or convenient. Furthermore, both men and women who had no companion to walk with along their walking routes including pets, were also less likely to walk for exercise or recreation. This observation was particularly more so for women than for men. This could also be due to safety reasons. Both men and women who were classified as having good or poor physical and mental health all felt that the aesthetic appeal of the environment and companionship played a prominent role in their decision to walk or not for exercise or recreation.

It was therefore concluded that convenience, environmental aesthetics and companionship are all significant determinants for walking and thus should be strongly considered when formulating policies. Indeed, if policy makers are to promote walking, they should to focus on ensuring that walking paths are smooth and convenient so that people can reach their destinations quickly, comfortably and safely. Furthermore, governments and municipalities should also invest in creating beautiful parks, planting trees and providing amenities such as public toilets, water fountains and benches that will all

contribute to a more aesthetically appealing environment.

As already suggested, technological equipment can help in encouraging walking amongst the public. Information and Communication Technology (ICT) can provide incentives for people to track their progress through counting number of steps and gamify the experience of walking, monitoring heart rate and calories burnt [19]. The SPN system [20] incorporates such features so that users will be more inclined to walk and thus improve health and sustainable development. In addition, SPN may have a way of informing users of the CO2 emissions that have been reduced as a result of their walking journeys. By entailing such features, SPN can be an important contributor to combating climate change. Papageorgiou et al [21], discuss how SPN will show tailored walking routes to users and enable them to enjoy their walks with family and friends as well as collect rewards for engaging in active mobility. Moreover, SPN acts as a channel by which SPN users can contact and inform governments and municipalities about problems in current pedestrian conditions.

What becomes very important is to be able to justify investing in the implementation of a Smart Pedestrian Network Information system. Such a justification could be carried out by quantifying the benefits accrued from implementing SPN and promoting active mobility. In the next section, a framework is presented for evaluating such benefits in a quantified manner. Note that the proposed framework could be utilised by other similar projects in order to prove their viability. In this way, a significant gap is fulfilled in the current literature.

III. A FRAMEWORK FOR EVALUATING THE BENEFITS OF ACTIVE MOBILITY

Consider a case where the development team of SPN wants to target a specific share of the market denoted by x , within a given region. Let α be the population that lies within this specific region. Therefore, the targeted market denoted by y within the given region would be:

$$y = x * \alpha \quad (1)$$

Based on the literature review, it was revealed that walking can contribute a lot to lowering healthcare costs associated with obesity, cardiovascular diseases (CVDs) such as coronary artery disease, diabetes and deaths attributed to pollution. The number of people within the given population of the region α that suffer from obesity, coronary artery (CA) disease, diabetes and deaths attributed to pollution are characterized by β , γ , ϕ and λ respectively. To calculate the number of individuals suffering from obesity, CVDs, diabetes and deaths attributed to pollution within the target audience y which will be denoted by r , q , s and z , we estimate the following:

$$r = \left(y * \frac{\beta}{\alpha} \right) \quad (2)$$

$$q = \left(y * \frac{\gamma}{\alpha} \right) \quad (3)$$

$$s = \left(y * \frac{\phi}{\alpha} \right) \quad (4)$$

$$z = \left(y * \frac{\lambda}{\alpha} \right) \quad (5)$$

Data on the amount on healthcare costs pertaining to pollution within the region can be obtained. It is estimated that 75% of air pollution is caused by vehicle use which has significant negative effects on peoples' health [22]. Therefore, 75% of healthcare costs associated with air pollution can be reduced by increasing walking levels. Let these healthcare costs be denoted by h . To calculate the per capita healthcare costs c , the healthcare costs h should be divided by the total population α . Based on a study conducted by the WHO [23], the healthcare costs h associated with air pollution, represents 4% of GDP as shown in the equation below:

$$\text{€}c = \left(0.04 * \frac{GDP}{\alpha} \right) \quad (6)$$

Each person in the target region on average accounts for $\text{€}c$ in healthcare costs owed to pollution.

Let it be further assumed that implementing SPN can influence people to walk a further 20,000 steps on average per week which corresponds to 18km/week. Therefore, each of these individuals within the target audience who suffer from obesity, coronary artery disease, diabetes or carry the risk of death, can reduce this healthcare costs by a specified amount. A study [24] have found that walking can reduce cholesterol levels by 7% thereby reducing obesity, CA disease by 9% diabetes by 12% and deaths by 27%. Therefore, estimated reduced healthcare costs of obesity, CVDs, diabetes and deaths in the target audience will be:

$$ObesityInd = (c) * (Obesity) * (0.07) \quad (7)$$

$$CA diseaseInd = (c) * (CA disease) * (0.09) \quad (8)$$

$$DiabetesInd = (c) * (Diabetes) * (0.12) \quad (9)$$

$$DeathsInd = (c) * (Deaths) * (0.27) \quad (10)$$

Where $ObesityInd$, $CA diseaseInd$, $DiabetesInd$ and $DeathsInd$ are the healthcare cost savings realized from obesity, CVDs, diabetes, and death respectively if people who fall within these categories walk 20,000 more steps per week. The total cost savings in healthcare therefore is the total sum of these four variables:

$$\begin{aligned} HealthcareCostSavings = & ObesityInd \\ & + CA diseaseInd \\ & + DiabetesInd \\ & + DeathsInd \end{aligned} \quad (11)$$

In a similar fashion, we can estimate the possible cost savings that individuals make from reduced vehicle use. More specifically, if individuals walk more and use their cars less, they will be able to reduce fuel costs. On average in Europe, it was estimated that fuel prices are $\text{€}1.34/\text{liter}$ [25] and, on average, cars within the European Union travel 12,000 kilometers per year [26]. Since there are 52 weeks in a

calendar year, this equates to 230km/week. A fuel efficient car on average consumes 1 liter of fuel for every 16km. Therefore, if an individual walks 18km per week, they will save approximately $\text{€}1.50$ on weekly fuel. This can then be aggregated for all individuals using SPN so that the total amount saved on petrol in this region is $y * \text{€}1.50$. This can then be multiplied by 52 weeks, to estimate cost savings made on petrol per year.

SPN would also provide users with incentives to motivate them to walk. This can be in the form of free/discounted meals/drinks, discounted cinema tickets or even cash rewards. Whatever form the discounted/free item takes, this implies that an individual will have more money each month to spend elsewhere. Let it be assumed that each SPN user can save 5% of their monthly spending by redeeming rewards. Obtaining data on the average monthly spending in this specified region, will enable us to calculate the monthly savings accrued from SPN rewards and incentives per capita and for the entire target audience. If the monthly spending is b , then each SPN user can save $\text{€}b * 0.05$ every month to walk around 80,000 steps.

If the market share of SPN is to grow each year, the realized benefits will grow as well. To calculate the benefits, an appropriate discounted cash flow method should be applied. The framework makes use of Net Present Value (NPV) and suggests employing a constant discount rate that will be denoted by r . For a given period T that denotes the number of years, r can be applied to estimate the NPV of each of these benefits. Therefore, NPV is such that:

$$NPV = \frac{Benefits}{1+r} + \frac{Benefits}{(1+r)^2} + \frac{Benefits}{(1+r)^3} \dots + \frac{Benefits}{(1+r)^T} \quad (12)$$

Therefore, NPV can be written as:

$$NPV = \sum_{T=0}^n \frac{Benefits}{(1+r)^T} \quad (13)$$

The income streams of SPN can be calculated in a similar manner. SPN will be free to download with a Freemium and Premium option available. The Premium will be useful for those wishing to obtain more advanced SPN features such as expert advice from nutritionists or greater rewards from walking. SPN Premium will cost $\text{€}1/\text{month}$. Therefore, if it is assumed that 10% of SPN users will purchase the Premium version, this will amount to:

$$MonthlyPremiumIncome = \text{€}1 * 0.1 * y \quad (14)$$

Also, SPN will be open to donations. For instance, donations can be made of a specific amount, say $\text{€}0.50$ for every SPN user. This can amount to $\text{€}0.50 * y$ per year. Similarly, sponsorship are another way that the SPN project can raise funds. We can collaborate with firms interested in sustainability to promote their businesses in return for extra funding. So, companies might be especially interested in sponsoring SPN.

For instance, €1 every year for every SPN user can be given which will amount to €1 * y per year.

Furthermore, advertising will be another main source of SPN income. This will be via push button profits. For every advert that is clicked on by SPN users, SPN will gain €0.10. Let it be assumed that 5% of the targeted audience will click on adverts every day. This will amount to €0.10 * 0.05 * y . Multiplied by 365 days, and divided by 12 to calculate income from adverts for a calendar month, we obtain:

$$\text{MonthlyAdvertisementIncome} = \frac{(0.10 * 0.05 * y)(365)}{12} \quad (15)$$

An important source of income for SPN will arise from insurance partner companies or medical funds as these companies and funds will realize cost savings in terms of lower healthcare costs. As a result, a percentage of these cost savings should be paid back to the SPN users, with a percentage given to SPN. These amounts can be calculated based on the number of steps walked by SPN users. A viable proposition would be that 25% of cost savings will be kept by insurance companies, 25% will be given to SPN and 50% will be given to SPN users. Let us assume that 30% of insurance companies and/or medical funds within the specified region cooperate with SPN. We can calculate the income of SPN from healthcare cost savings per step for every year as follows:

$$\text{HealthcareIncome} = 0.25 * 0.30 * \text{HealthcareCost} \quad (16)$$

Further, support from municipalities and governments will also provide additional sources of income for SPN. As already seen, the cost savings of such an application can be significant. Local authorities can use these funds towards other government projects. Such government projects can include supporting SPN as a means of promoting sustainable development. A small fraction of the cost savings that make due to increased levels of walking can be given to SPN. *NPV* can also be calculated for income streams and costs of SPN, by applying the constant discount factor r such that the total *NPV* of the SPN project will be:

$$\begin{aligned} \text{TotalNPV} &= \text{NPV of Income Streams} - \text{NPV of Costs} \\ &= \sum_{T=0}^n \frac{\text{Income}}{(1+r)^T} - \sum_{T=0}^n \frac{\text{Costs}}{(1+r)^T} \quad (17) \end{aligned}$$

The costs of implementing SPN per city are relatively low, since the software of SPN has already been developed via a European funded project (sunk cost). It is estimated that the monthly cost for the server is €200/month while the cost of supporting SPN via Apple and Google app stores is €90/month. In addition, we will have personnel, marketing and advertising costs, as well as overheads. Running a hypothetical scenario using computer simulation, it became apparent that SPN is a worthwhile investment. Comparing the

costs with the revenue streams, it is clear that implementing the SPN project is indeed a profitable venture.

Note that there are several other benefits that have not been accounted for in the proposed framework such as higher productivity, utility from walking, social capital levels, and reduced building maintenance costs from air pollution. If these were also accounted for, definitely, benefits would be higher.

IV. CONCLUSION

Prior to investing in Information and Communication Technology (ICT) solutions for promoting sustainability it is important to understand the benefits realized by governments, individuals, the economy, the natural environment and generally, our planet. To provide clarity, quantification of such benefits becomes essential. However, the process of quantification is far from simple, especially for the case of investing in ICT to promote active mobility. For instance, when a person walks, they obtain a certain level of satisfaction or utility. But how can we measure accurately this level of utility? There is no consensus on an appropriate method to calculate utility. Often, surveys are given out to people for them to rank or rate their preferences. Furthermore, happiness indexes are used to be able to gather information on utility levels. However, this implies relying on information delivered by individuals themselves regarding their utility evaluation. But often, people incorrectly estimate their true utility levels. People have a tendency to also overestimate the utility from tangible items such as discounted/free goods they can get from using SPN. In contrast, people tend to underestimate the utility they obtain from non-tangible items. In our case, this can involve their increased social capital levels and utility.

It is evident that placing a monetary value on something that has no intrinsic value is far from simple. On the other hand, by gathering as much information as possible from people and perhaps implementing other methods such as looking at a persons willingness to pay for something, may be a good first step to build a solid grasp on how people view active mobility.

In this paper, we evaluated benefits of implementing a Smart Pedestrian Network (SPN) information system which promotes walking. This is achieved by developing a framework for quantifying the main benefits that arise from using the SPN system and increased walking. It can be clearly seen that walking benefits the natural environment, people, governments, the economy and ecosystems. Walking can reduce the likelihood of developing certain health issues including diabetes, obesity, cardiovascular diseases and even premature death. Consequently, it can help to significantly reduce healthcare costs. In return, this not only helps stimulate greater economic activity and development by promoting greater labor productivity, but it also enables governments and people to save on healthcare costs and allocate funds to other projects or needs they may have. Similarly, if people save on fuel costs, they will see an increase in their real income and their monthly budgets.

Implementing SPN would also encourage people to walk more by providing incentives in the form of rewards. There-

fore, people can also reduce monthly expenses by walking as they receive free/discounted goods, vouchers for shopping and even cash rewards. Calculating the costs and income streams showed that SPN has great viability as the revenue streams can be relatively large, and thus would be a worthy investment. All costs and income streams can be calculated in the proposed framework using discounted cash flow methods. It is clear that implementing sustainable Information and Communication Technology (ICT) solutions such as the SPN system produce long term benefits that will help contribute to a more sustainable planet.

ACKNOWLEDGMENT

The research presented in this paper is co-funded by the Republic of Cyprus and the European Regional Development Fund as part of ERANET Cofund Smart Urban Futures (ENSUF) Joint Programming Initiative (JPI) Urban Europe, through the Research Promotion Foundation, protocol no. KOINA/IIKII URBAN EUROPE/1215/11. This framework is supported by the European Commission and funded under the HORIZON 2020 ERANET Co-fund scheme

REFERENCES

- [1] G. Papageorgiou and A. Maimaris, "Towards the development of Intelligent Pedestrian Mobility Systems (IPMS)," in *2017 International Conference on Electrical Engineering and Informatics (ICELTICs)*, Oct 2017, pp. 251–256.
- [2] D. Berthier. (2018, Nov.) Air pollution kills more than 500,000 Europeans each year. European Data Journalism Network. Accessed: 2019-09-01. [Online]. Available: <https://www.europeandatajournalism.eu/News/Data-news/Air-pollution-kills-more-than-500-000-Europeans-each-year>
- [3] The World Bank. (2016, Sep.) Air Pollution Deaths Cost Global Economy US\$225 Billion. Accessed: 2019-09-01. [Online]. Available: <https://www.worldbank.org/en/news/press-release/2016/09/08/air-pollution-deaths-cost-global-economy-225-billion>
- [4] J. He, H. Liu, and A. Salvo, "Severe Air Pollution and Labor Productivity: Evidence from Industrial Towns in China," *American Economic Journal: Applied Economics*, vol. 11, no. 1, pp. 173–201, January 2019. [Online]. Available: <http://www.aeaweb.org/articles?id=10.1257/app.20170286>
- [5] M. Pal, M. Yirgalem, M. Anberber, B. Giro, and R. Dasgupta, "Impact of Environmental Pollution on Animal Health," *Journal of Natural History*, vol. 11, no. 1, pp. 1–3, Jun. 2015.
- [6] S. Rogers, K. Gardner, and C. Carlson, "Social capital and walkability as social aspects of sustainability," *Sustainability*, vol. 5, no. 8, pp. 3473–3483, 2013.
- [7] J. Barton, R. Hine, and J. Pretty, "The health benefits of walking in greenspaces of high natural and heritage value," *Journal of Integrative Environmental Sciences*, vol. 6, no. 4, pp. 261–278, 2009. [Online]. Available: <https://doi.org/10.1080/19438150903378425>
- [8] A. Wattanapit and S. Thanamee, "Evidence behind 10,000 steps walking," *Journal of Health Research*, vol. 31, no. 3, pp. 241–248, 2017.
- [9] WHO Regional Office for Europe. (2014) Data and statistics: The challenge of obesity - quick statistics. Accessed: 2019-09-01. [Online]. Available: <http://www.euro.who.int/en/health-topics/noncommunicable-diseases/obesity/data-and-statistics>
- [10] B. Senauer and M. Gemma, "Why is the obesity rate so low in Japan and high in the US? Some possible economic explanations," *Tech. Rep.*, 2006.
- [11] Harvard Medical School. (2019) 5 surprising benefits of walking. Harvard Medical School. Accessed: 2019-09-01. [Online]. Available: <https://www.health.harvard.edu/staying-healthy/5-surprising-benefits-of-walking>
- [12] J. O. Hill, "Walking and Type 2 Diabetes," *Diabetes Care*, vol. 28, no. 6, pp. 1524–1525, 2005. [Online]. Available: <https://care.diabetesjournals.org/content/28/6/1524>
- [13] WHO Regional Office for Europe. (2014) Data and statistics: The challenge of diabetes. Accessed: 2019-09-01. [Online]. Available: <http://www.euro.who.int/en/health-topics/noncommunicable-diseases/diabetes/data-and-statistics>
- [14] C. Loong, D. van Lierop, and A. El-Geneidy, "On time and ready to go: An analysis of commuters punctuality and energy levels at work or school," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 45, pp. 1 – 13, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S136984781630571X>
- [15] H. Kim and S. Yang, "Neighborhood Walking and Social Capital: The Correlation between Walking Experience and Individual Perception of Social Capital," *Sustainability*, vol. 9, no. 5, 2017. [Online]. Available: <https://www.mdpi.com/2071-1050/9/5/680>
- [16] Harvard Kennedy School. (2018) SocialCapitalShort-FormSurvey. Harvard Kennedy School. Accessed: 2019-05-01. [Online]. Available: <http://www.hks.harvard.edu/programs/saguaro/measurement#shortform>
- [17] G. Papageorgiou, A. Maimaris, T. Efstathiadou, and E. Balamou, "EVALUATING ATTITUDES ON THE QUALITY OF SERVICE OF PEDESTRIAN NETWORKS," in *WIT Transactions on The Built Environment*, S. Ricci and C. A. Brebbia, Eds., vol. 176. WIT Press, sep 2017, pp. 35–41.
- [18] K. Ball, A. Bauman, E. Leslie, and N. Owen, "Perceived Environmental Aesthetics and Convenience and Company Are Associated with Walking for Exercise among Australian Adults," *Preventive Medicine*, vol. 33, no. 5, pp. 434 – 440, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0091743501909120>
- [19] E. Conticelli, A. Maimaris, G. Papageorgiou, and S. Tondelli, *Planning and Designing Walkable Cities: A Smart Approach*. Cham: Springer International Publishing, 2018, pp. 251–269. [Online]. Available: https://doi.org/10.1007/978-3-319-77682-8_15
- [20] G. Papageorgiou, T. Efstathiadou, A. Efstathiades, and A. Maimaris, "Promoting Active Transportation via Information and Communication Technologies," in *IEEE EUROCON 2019 - 18th International Conference on Smart Technologies*, July 2019, pp. 1–5.
- [21] G. Papageorgiou, D. Demetriou, E. Balamou, and A. Maimaris, "Market research and concept study for a smart pedestrian network application," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, ser. PETRA '18. New York, NY, USA: ACM, 2018, pp. 522–528. [Online]. Available: <http://doi.acm.org/10.1145/3197768.3201550>
- [22] European Environment Agency. (2016, Apr.) Premature deaths attributable to air pollution. European Environment Agency. Accessed: 2019-09-01. [Online]. Available: <https://www.eea.europa.eu/media/newsreleases/many-europeans-still-exposed-to-air-pollution-2015/premature-deaths-attributable-to-air-pollution>
- [23] World Health Organization. (2019) How air pollution is destroying our health. Accessed: 2019-09-01. [Online]. Available: <https://www.who.int/air-pollution/news-and-events/how-air-pollution-is-destroying-our-health>
- [24] R. Glatter. (2013, Apr.) Brisk Walking or Running: Both Reduce Risk for Heart Disease. Forbes. [Online]. Available: <https://www.forbes.com/sites/robertglatter/2013/04/05/brisk-walking-or-running-both-reduce-risk-for-heart-disease/#5281bd61461e>
- [25] Drive Alive. (2019, Jul.) Fuel prices in Europe: the cost of petrol and diesel in Europe. All petrol and diesel prices in Euros at July 18th 2019. Accessed: 2019-09-01. [Online]. Available: https://www.drive-alive.co.uk/fuel_prices_europe.html
- [26] Odyssee-Mure. (2019) CHANGE IN DISTANCE TRAVELLED BY CAR. Accessed: 2019-09-01. [Online]. Available: <https://www.odyssee-mure.eu/publications/efficiency-by-sector/transport/distance-travelled-by-car.html>

Real-time processing of cybersecurity system data for attacker profiling

1st Patrik Pekarčík2nd Tomáš Kekeňák3rd Pavol Sokol4th Terézia Mézešová*Computer science institute**Computer science institute**Computer science institute**Computer science institute**Pavol Jozef Safarik University Pavol Jozef Safarik University Pavol Jozef Safarik University Pavol Jozef Safarik University*

Košice, Slovakia

Košice, Slovakia

Košice, Slovakia

Košice, Slovakia

patrik.pekarcik@student.upjs.sk tomas.kekenak@student.upjs.sk

pavol.sokol@upjs.sk

terezia.mezesova@upjs.sk

Abstract—Usage of cybersecurity tools entails an enormous amount of data that brings the possibility of different approaches to the processing of cybersecurity data. This paper discusses the profiling of attackers, which, in practice, can help in managing cybersecurity events. The main goal of the research is to perform attackers' profiling as close as possible to real-time processing. The paper outlines the basic idea of real-time attacker profiling. We use stream processing. Within the system, we profile attackers into seven profiles or mark them as outliers if they do not fall into any of the known profiles. The paper also deals with the dynamic profiling model update and the difference of the calculated model using the original non-real-time model.

Index Terms—attacker profiling, real-time stream processing, classification, cybersecurity data

I. INTRODUCTION

As the number of heterogeneous devices in computer networks increases, the number of security incidents that security analysts must address is also increasing. Among the standard devices we encounter in the current computer networks, we can include mobile phones, Internet of Things (IoT) devices, such as smart coffee machines, valves, locks. Host-based defence solutions, such as antivirus, are impractical for these devices due to their high consumption of resources. For this reason, network security specialises in monitoring network traffic, tracking application logs from specific devices, network devices or network services. We can name these solutions as passive because they do not limit the work of the devices. Since we are dealing with monitoring network traffic, we have to think about a large amount of data and look for a solution which will be very responsive to changes in network security data.

All the mentioned data is flowing continuously to the central security unit. In this type of big data application, it is not necessary to process entire data at once. Most of big data applications are just streaming current data to processing units [1]. This type of processing is called Stream Processing. It allows applications to efficiently exploit a limited form of parallel processing, without explicitly managing allocation, synchronisation or communication among these units [2].

This research is funded by the VVGS projects under contract No. VVGS-PF-2019-1062, VEGA project under contract No. VEGA/A-1/0056/18, and Slovak Research and development agency (SRDA) project under contract No. APVV-17-0561.

This paper aims to design and implement a real-time classification of a threat. Threat profiling consists of extracting behaviour characteristics of detected threats and clustering them into distinct groups called profiles and subsequently classifying any incoming threats into the predefined profiles. To achieve this aim, we build on the research of attacker profiling in [3]. Mentioned research is comparing several methods for creating clusters of a threat. They found that partitioning around medoids (PAM) clustering method will act with good results. Also, they reasonably discuss the number of searching clusters and seven clusters acted with cleaner results - internal measures and stability measure in combination with external facts indicated seven as an appropriate number of clusters. The problem identified in the research is that the potential attack is revealed with a considerable time delay. A status alert that includes two-week data is not as relevant as an alert about current activity. We extend the profile of attackers used in this way in order to classify attackers in real-time using a streaming approach. The principle of current processing is data stream processing and verification of fulfilment of conditions of set computational models. This type of processing performs the data calculations within a short time after receiving the data. Usually, it takes from milliseconds to minutes. Profiles adapting to the new incoming threats in real-time is an active research area proposed in the reviewed literature. Based on the above, we state the following research sub-goals:

- 1) design a model for real-time profiling, and
- 2) design and implement a system for real-time profiling.

This paper is organised into five sections. Section II focuses on the review of the published papers on profiling and related topics. Section III outlines the dataset. Section IV focuses on the design and implementation of a system for real-time profiling. In Section V, we outline the model of real-time profiling, including aggregation, classification, model actualisation and results. The last section contains conclusions and suggestions for the future research.

II. STATE OF ART

Usage of the k-means clustering method on network data was proposed in [4] to create various activity groups and thus identify any abnormal activity on the network. Wang, Hu and Hu surveyed the approaches towards profiling the behaviour

of network traffic of target hosts in [5]. They presented various techniques which differ in the kind of network data that is used to extract the profile defining features. They identified profiling based on multi-source information and lack of consensus on evaluation metrics as still open issues. Network backbone traffic was profiled to detect anomalies in [6]. Over 40 attributes characterised the profiles. Ant colony optimisation algorithm was used to find the anomalies. Their experimental results showed an accuracy of 97.4% and false positive rate of 0.9%. In [7], communication between hosts in a backbone network was modelled as a bipartite graph and clustering are performed based on the closeness of the nodes, defined by a clustering coefficient which can be applied both locally or globally. It is an adapted, simple spectral clustering algorithm. There were also described the practical benefits of exploring the behaviour of network end-hosts, such as detecting scanning activities, early phases of worms, distributed DoS attacks. Hammerschmidt et al. proposed an approach in [8] towards learning and distinguishing among various profiles in network communication which requires a small number of attributes. Probabilistic deterministic finite automata were adapted and required less training data. The objective of the profiling module is to distinguish between legitimate and botnet traffic. In [9], the authors presented a pipeline where they can identify and cluster together typical attack profiles, focusing on detection of distributed reflective DoS attacks. They identified 13 clusters. Various clustering algorithms were tested (k-means, self-organising maps, expectation maximisation, and others). DBSCAN algorithm identified in [10] 5 profiles of attack behaviour - scanning a single port, port scanning on a single host, server traffic behaviour, clients sending HTTP-like requests, P2P traffic. In [11], k-means clustering was used for Netflow records. Their results were 4 clusters, each characterised by nine attributes.

The reviewed related works focus on the creation of the model in an offline mode, where all the learning data is available, and no new data is incoming. In their future works section, they mention that the next stage of the research is performing the techniques in real-time. We describe our approach to it in this paper.

III. DATASET

The source of data for our research is the alerts obtained from a Warden system [12]. It is a system that supports sharing information about security events on individual computer networks connected to this system. Data is stored and shared in IDEA format (Intrusion Detection Extensible Alert) [13]. IDEA format is a descriptive data model using a key-value JSON structure. The main detection sources of data that send IDEA alerts to the Warden system can include attack detection systems, honeypots, network flow probes, system log records. Alert in the IDEA format contains several mandatory attributes (format, ID, detect time, category) [14] and many optional fields with multiple input support. The fields we follow most in this research are the category, network traffic source data (IP address, port, protocol), network traffic target data (IP,

port, protocol), detection time, and interruption time. For this research, data were collected during four weeks (from 2018-01-12 to 2018-02-09) by the Warden system. We have split security data into four parts per each week. Collected data contain approximately 216 million records from various data sources mentioned above.

In Tab. I it is shown the specific attributes of each analysis week, especially number of IP address, number of the Internet service providers, average of security incident category - recon scanning.

TABLE I
INFORMATION ABOUT WEEKS IN DATASET

Week	No. IP	No. ISP	Avg of Cat. Recon-scanning
Week 1	546 359	16 004	9.33
Week 2	528 982	15 832	10.18
Week 3	497 512	15 915	10.92
Week 4	520 800	15 887	11.00

IV. DESIGN AND IMPLEMENTATION OF A SYSTEM FOR REAL-TIME PROFILING

We now present the input data and the way of their current aggregation and subsequent classification. We describe the processing of security data in individual processing steps. We have verified this approach using the Python scripts with the usage of PostgreSQL memory database. The following diagram (Fig. 1) shows how we designed the profiling of attackers by using security data stream processing. It is a module to simplify the definition of individual processing steps and communication between them. The schema contains separate processing steps to receive events continually. The events are then processed according to the current processing step and responded by sending the processed data to the next processing step. Some processing steps are illustrated with pseudocode for a better understanding of processing complexity at them.

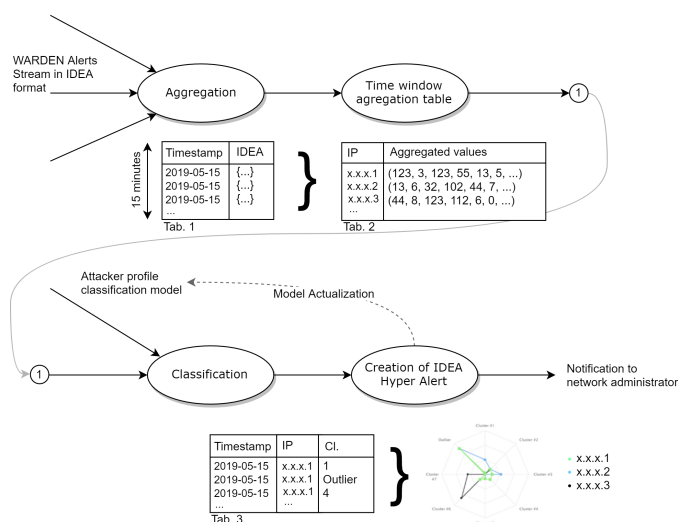


Fig. 1. Security data flow stream processing scheme, to network administrator notification.

After classification, we obtain a table of attackers assigned to clusters. The data of the different classifications over time are aggregated and visualised in the spider graph shown in Fig. 2 according to the number of classifications to the attacker's profile. In the visualisation, we can observe how the behaviour of the attacker had changed in the monitored time (for example, 30 minutes before the attack was detected). The most prominent figure in the monitored chart is membership in the group of the outliers. The main reason for this high frequency is the early stages of scanning our IP addresses during which we still do not have enough data.

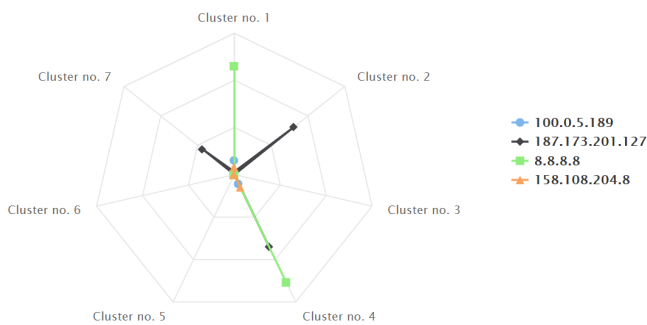


Fig. 2. Visualization of Cluster Allocation by IP Address.

A. Implementation

To test our methodology of stream processing cybersecurity data, we have written the application in Python version 2.7. We also used the PostgreSQL memory database for storing current window data. Our application has separated each processing steps shown in Fig. 1. For testing reasons, we do not use streaming input from a real network. Our entire data from 4 weeks was stored in a single table with rows representing single IDEA alert, and every row had a timestamp of receive. Doing it like this, we could process data faster as it would be in production. In our test environment which is Linux kernel 4.19.0-5-amd64 on Intel(R) Xeon(R) E5-2620 v4 @ 2.10GHz with 128GB RAM and SSD drives. We were testing in two separated conditions. First was without dynamic model recalculation, and the second one was recalculating medoid after every classification. We managed to process each 1 minute of latest window IDEA events (it is about 20000 events) to process under 10 seconds for the first condition, and in the second condition, it took 15 - 20 seconds depending on the amount of data in the current window. With this information, we are preparing the environment for production environment for stream processing.

V. MODEL FOR REAL-TIME PROFILING

In this section, the model for real-time processing of network events and assigning them to a particular profile is described.

A. Aggregation

The purpose of this module is to receive IDEA alerts and locally create a set of alerts for a specified time. In our experiment, we retained data for the last 15 minutes. An analysis of data collected from the Warden system is difficult without their transformation. For this reason, they had to be preprocessed. Each record from the Warden stands for a security event. In the context of this paper, an attacker, or a threat agent is a specific system entity with a public IP address or several system entities of the same private network subnet using that public IP address to communicate with other devices on the Internet (e.g. using NAT) and perform a threat action. From this data, a table in Fig. 1) with 12 columns were made by transforming data. Each column has its data type. Therefore, it is easier to perform specific operations, for example, numerical operations which were not possible to do directly from the JSON format. Columns in Tab. II represent following properties: (I) ID; (II) source IP address; (III) target IP address; (IV) category; (V) category count; (VI) protocol; (VII) protocol count; (VIII) port; (IX) duration; (X) start timestamp; (XI) end timestamp; and (XII) Internet service provider (ISP).

However, this table contains attacks, not threat agents; therefore, another transformation was needed. This transformation consists of merging the same source IP addresses, thus creating one entry per one threat agent.

TABLE II
TRANSFORMED IDEA JSON DATA TO TABLE

Order	Column title	Example value
1.	ID	12345
2.	Source IP	10.158.x.10
3.	Target IP	10.159.x.11
4.	Category	Recon.Scanning
5.	Category Count	10
6.	Protocol	TCP
7.	Protocol Count	1
8.	Port	80
9.	Duration	600s
10.	Start Timestamp	2018-03-18 17:45:15
11.	End Timestamp	2018-03-18 17:55:15
12.	ISP	AS100000

In the final output of this processing step, every threat agent is represented by a 7-dimensional vector. Vectors representing threat agents consist of the following attributes: count of category Recon.Scanning, count of category Availability.DDoS, duration, maximal idleness, minimal idleness, ISP, and count of unique targets. We store the aggregated data in the form of vectors, as shown in Tab. III. The content that was updated in this table within this processing step will be forwarded to the next processing step for further processing. Processing in this step is described in Algorithm 1.

B. Classification in real-time

The classification module needs two inputs. The first input is a classification model for profiling attackers. We used a model that was previously calculated from two-week collected

TABLE III
AGGREGATED DATA TABLE

Order	Column title	Example value
1.	Source IP	10.158.197.10
2.	Recon.Scanning Count	156
3.	Availability.DDoS Count	155
4.	Duration	685s
5.	Maximal Idleness	80s
6.	Minimal Idleness	60s
7.	ISP count	10
8.	Unique Targets Count	254

Algorithm 1 Events aggregation and sending to next step

```

Select
  IP,
  Count(CategoryReconScanning),
  Count(CategoryAvailabilityDDoS),
  Sum(Duration),
  Max(IdleTime),
  Min(IdleTime),
  ISP,
  Count(Distinct Targets)
From TransformedIDEASecurityData
Group By IP

# Send Threat Agent Vectors to next step
For (Let Row in Select)
  Send Row As ThreatAgentVector

```

security data based on the algorithm from paper [3]. In this paper we used partitioning around medoids (PAM) clustering method [15] to find representative objects (medoids among the observations of the dataset) of clusters which minimise the sum of the dissimilarities of the observations to their closest representative object. A medoid (centre) is a representative of a cluster, chosen as its most central object. Their model contains the medoids of the seven clusters stored in the vector. This module contains the ranges of attributes in which we can confidently classify future attackers. The second input for this module is the data stream coming from the data preprocessing aggregation. This stream contains IP addresses of their respective vectors of aggregated defined attributes.

Based on the input data mentioned above, this stream-receiving module classifies the received IP address vector into one of the models defined clusters. If the vector is not included in any of the clusters, then the vector is included in the outlier group. The outlier group represents the attackers whose behaviour is in a completely different way than most attackers. The output of this module is a classified IP address and the creation of a correlated hyperalert enriched with the profile of the attacker where it was included (See Tab. IV). Processing in this step is described in Algorithm 2. This code classifies in $O(k)$ where k represents the count of clusters.

TABLE IV
CORRELATION OF ASSIGNED PROFILES FOR SOURCE IPs

Timestamp	Source IP	Assigned to
2018-03-18 17:45:00	10.158.197.10	Cluster 5
2018-03-18 17:45:00	10.199.197.10	Outlier

Algorithm 2 Finding minimal vector distance to cluster model center

```

Receive ThreatAgentVector
Load ClusteringModels
# |ClusteringModels| = 7

Let MinDistance = Infinity
Let MinVector = Outlier
For(Let ClusterName, ClusterVector In
  ↪ ClusteringModels)
  Set Distance =
    ↪ FindDistanceBetweenVectors(
    ↪ ClusterVector,
    ↪ ThreatAgentVector)
  If(Distance != null OR Distance <
    ↪ MinDistance)
    MinDistance = Distance
    MinVector = ClusterName
Send MinVector As ThreatAgentClusterName
Send ThreatAgentVector As
  ↪ ThreatAgentVector

```

C. Classification model actualization

The current processing solution proposed so far has one disadvantage. In the beginning, the calculated model from the previous paper was calculated on a two-week data sample, and it is recommended to update clustering at regular intervals to keep the application up to date. However, such a recalculation has nothing to do with stream processing, and it would be an external input to the proposed solution for us. That is the reason why we have also focused on dynamic model updates. We have modified the algorithm so that after successful classification of a new record it could update the centre of the cluster to which it was assigned. We have used the weighted average to recalculate the centre of the cluster. The model automatically updated for two weeks was compared to a statically calculated model according to [3]. An important observation was that the dynamic update did not work towards the same model. The main reason why the models were different over the same period and the same security data is the following: in stream processing, model updates on partially received aggregated records. That is the reason why one threat agent can update the cluster model multiple times to the end of his activity. Processing in this step is described in Algorithm 3.

After running the number of PAM iterations on different data, we figured out that it is needed to name each calculated

Algorithm 3 Finding minimal vector distance to cluster model center

```

Receive ThreatAgentVector
Receive ThreatAgentClusterName
Load ClusteringModels

If(ThreatAgentVector != Outlier)
  Let ThreatAgentClusterVector =
    ↳ ClusteringModel.Get(
    ↳ ThreatAgentClusterName)
  NewThreatAgentClusterVector =
    ↳ ThreatAgentClusterVector.
    ↳ RecalcWithWeightAverage(
    ↳ ThreatAgentVector)
  ClusteringModels.Change(
    ↳ ThreatAgentClusterName,
    ↳ NewThreatAgentClusterVector)
Save ClusteringModels
  
```

cluster. The main problem began when the clustering model was recalculated, and the number of clusters was exchanged between them. We added an algorithm which classified each cluster from a new model. Then this algorithm assigned the same numbers to closest clusters. This cluster ordering helped us to stabilise clusters and made them more precise in time.

D. Results

Within this chapter, we have focused more closely on the analysis of cluster development over four weeks. In Fig. 3, we can see input data for creating threat agent models. The total number of events (green) and the total number of attacks (yellow) can give us a general picture of the overall state of security events in our dataset.

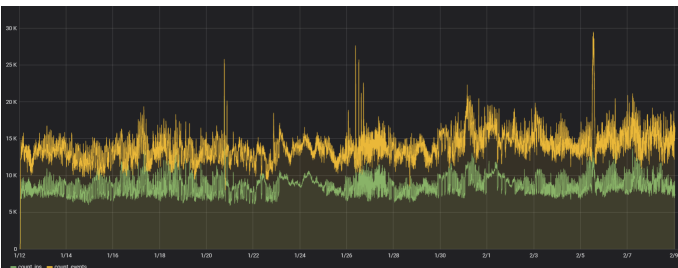


Fig. 3. Total number of security events (yellow) and IP addresses (green) over a period of 4 weeks.

Following figures 4 and 5 represents distribution of unique IP addresses in time window to the clusters. In Fig. 4 we can see that each cluster oscillates around a specific value, but several local maxima can be observed. These are not caused by a higher number of IP addresses, but by a change in the distribution of individual threat agent profiles.

The percentage distribution of the individual profiles within four weeks is shown in Fig. 4. The most significant changes

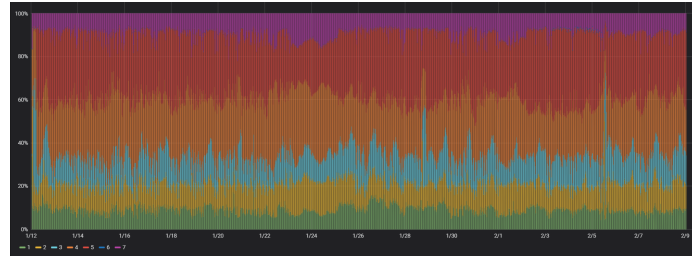


Fig. 4. Percentage distribution of clusters over a period of 4 weeks.

can be seen in cluster no. 3 (turquoise). Following figure Fig. 5 is zoomed into date of January 28, 2018, from 6:30 pm to 9:40 pm. The threat agent is characterised by, on average, seven different attack targets with a low span between individual security incidents.

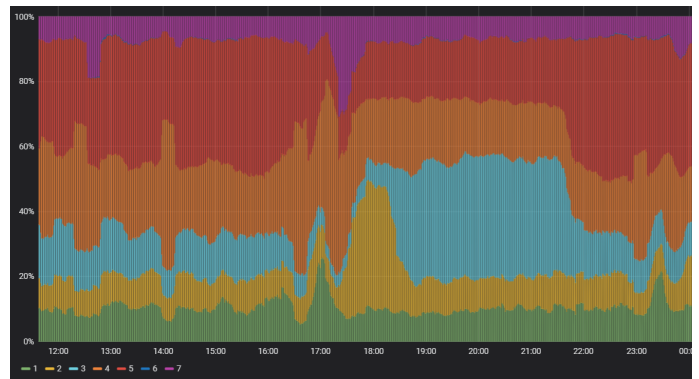


Fig. 5. Percentage distribution of clusters on 28.1.2019 from 12:00 to 24:00

VI. CONCLUSION AND FUTURE WORKS

In this paper, we discussed streaming applications to detect attackers in a computer network. We use security data from the Warden system as input security data. We have proposed a procedure for aggregation and classification these data by stream processing. Since only the changed data is transferred from one step to another in the streaming process, we can minimise the complexity of the calculations. We can also see that the modules are relatively simple for processor power. This results in a significant reduction in detection from the original two-week reports for immediate attacker profiling, as well as alerts to network administrators at the first hints of an attack in or towards their computer network. This is a big difference comparing to previous research.

Other questions that have arisen in the design of this solution include the analysis of how large of a time window can be used (currently 15 minutes). Another question for future research is the ability to reduce the weight of older activity to increase the time window to one hour. It is also advisable to check whether setting the minimum values (threshold) in each aggregated attribute helps reduce false positives when creating hyper alerts at the time of classification. Also, the opened question is whether vectors that do not pass the filter can affect

the dynamic modification of the model. Smooth transition of the attackers between the profiles raised the question of whether we can evaluate the strength in individual profiles.

ACKNOWLEDGMENT

This research is funded by project No. VVGS-PF-2019-1062, project No. VEGA/A-1/0056/18, and Slovak Research and development agency (APVV) project under contract No. APVV-17-0561.

REFERENCES

- [1] E. Mizell and R. Biery, "Introduction to gpus for data analytics," 2017.
- [2] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Computing Surveys (CSUR)*, vol. 44, no. 3, p. 15, 2012.
- [3] T. Bajtoš, A. Gajdoš, L. Kleinová, K. Lučivjanská, and P. Sokol, "Network intrusion detection with threat agent profiling," *Security and Communication Networks*, vol. 2018, 2018.
- [4] D. J. Marchette, "A statistical method for profiling network traffic." in *Workshop on Intrusion Detection and Network Monitoring*, 1999, pp. 119–128.
- [5] M. Wang, H. Hu, and G. Hu, "A survey on traffic-behavioral profiling of network end-target," in *Proceedings of the ACM Turing Celebration Conference-China*. ACM, 2019, p. 125.
- [6] X.-D. Zang, J. Gong, and X.-Y. Hu, "An adaptive profile-based approach for detecting anomalous traffic in backbone," *IEEE Access*, vol. 7, pp. 56 920–56 934, 2019.
- [7] K. Xu, F. Wang, and L. Gu, "Behavior analysis of internet traffic via bipartite graphs and one-mode projections," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 3, pp. 931–942, 2014.
- [8] C. Hammerschmidt, S. Marchal, R. State, G. Pellegrino, and S. Verwer, "Efficient learning of communication profiles from ip flow records," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, pp. 559–562.
- [9] L. Berti-Equille and Y. Zhauniarovich, "Profiling drdos attacks with data analytics pipeline," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1983–1986.
- [10] A. Jakalan, J. Gong, and S. Liu, "Profiling ip hosts based on traffic behavior," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2015, pp. 105–111.
- [11] T. Bakhshi and B. Ghita, "User traffic profiling," in *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015, pp. 91–97.
- [12] P. Kácha, M. Kostelec, and A. Kropacova, "Warden 3: Security event exchange redesign," in *Proceedings of the 19th International Conference on Computers: Recent Advances in Computer Science*, 2015.
- [13] P. Kácha, "Idea: security event taxonomy mapping," in *Proceedings of the 18th International Conference on Circuits, Systems, Communications and Computers*, 2014.
- [14] P. Kacha, "Idea: designing the data model for security event exchange," in *Proceedings of the 17th International Conference on Computers: Recent Advances in Computer Science*, 2013.
- [15] S. Tufféry, *Data mining and statistics for decision making*. John Wiley & Sons, 2011.

A Chatbot based on Deep Neural Network and Public Cloud Services with TJBOT Interface

1st Michal Povinský
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
antihutka@gmail.com

2nd Miroslav Melicherčík
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
miroslav.melichercik@umb.sk

3rd Vladimír Siládi
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
vladimir.siladi@umb.sk

Abstract—This paper describes an adaptation of the IBM TJBOT as a chatbot interface. TJBOT is an open source project designed to use artificial intelligence services in user friendly way. Primary it has been developed to be used with IBM Watson services. The adaptation has been done in three steps. In the first step deep neural network (DNN) based chat has been designed. Subsequently, three DNN were designed to perform experiments. They differed in training sets. The second step presents a join of DNN based chat with IBM Watson services (Speech-To-Text, Text-To-Speech). Originally, IBM Watson provides these services for limited number of languages. Slovak language is not included, too. Google cloud services fill this gap quite in good manner. This led to replacement of the IBM Watson services by the Google services. Finally, the chatbot is able to communicate in multiple languages, Slovak language including. Any non-English conversation has to be translated to English language and vice-versa by the Google translate service. This modified chatbot was tested by chat with randomly selected users.

Index Terms—deep neural network, deep learning, cloud services, TJBOT, Raspberry Pi, Chatbot

I. INTRODUCTION

TJBOT is an open source DIY project designed by IBM to help use IBM Watson artificial intelligence services in a popular way. “TJBOT was created by Maryam Ashoori at IBM Research as experiment to find the best practices in the design and implementation of cognitive objects” and promoted on November 9, 2016 [1], [2].

It can be used in two ways: to learn about cognitive systems and to build yourself an “intelligent” device based on cheap components (Raspberry Pi 3, RGB LED, Raspberry Pi camera, servomotor, speaker and microphone). The TJBOT kit contains a laser cut cardboard body. As researchers at IBM TJ Watson Research Center describe, “TJBOT can be programmed to listen, speak, see and recognize, shine its LED, understand emotions, and wave his arms” [3].

IBM researchers have developed three recipes to embody a cognition and this kit:

- make TJBOT respond to emotions,
- use user voice to control TJBOT,
- chat with TJBOT [1].

This work was supported by IT Academy - Education for the 21st Century ITMS 312011F057 and the Slovak Grant Agency KEPA contract No. 011UMB-4/2017.

In the Slovak Republic and the Czech Republic it is possible to buy at shops a Czech mutation of IBM TJBOT A Watson Maker Kit, designated as TJBOT CZ Edition. We have available this CZ Edition of TJBOT for our experiments. The leaders of CZ Edition project are J. Petřík (development) and J. Louda (project). The project blog (webpages) creates a space to knowledge exchange by the TJBOT CZ Edition community [4]. Their GitHub repositories offer: manuals to build TJBOT and bring it to life, simple starter program for TJBOTCZ, source files to lasercut or 3D print TJBOT body and library for TJBOTCZ projects [5].

The original TJBOT and TJBOT CZ Edition differs in some components used, especially in the diode and speaker. The CZ Edition contains KY-016 RGB LED and Anker nano speaker. The original kit contains NeoPixel Diffused 8mm Through-Hole LED and Mini External USB 2.0 Speaker. The original version of TJBOT had a problem with interoperability of the NeoPixel LED and audio jack output, which is caused by the limited number of PWM outputs available on Raspberry PI 3. When audio jack output is used, no more PWM pins are left to control the NeoPixel LED and vice versa [6]. Therefore they cannot be used simultaneously and when the NeoPixel LED is used, the audio jack output is blocked. In the CZ Edition the NeoPixel LED was replaced by a RGB LED controlled by 3 pins without need of PWM and audio jack output can be used for speaker. It led to make some changes in TJBOT CZ Edition’s software. At that time developers of IBM Watson made changes in the concepts of entering of authentication credentials into IBM Watson services (see Fig. 1). This time both editions of TJBOT use the latest method of the key authentication (see Listing 1).

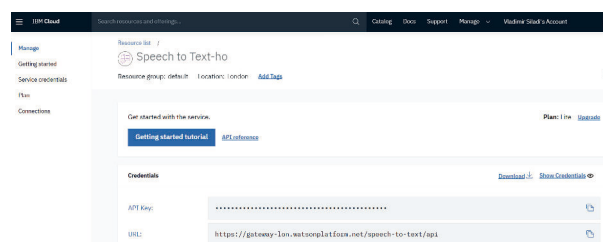


Fig. 1. IBM Watson – Speech to Text service credential

The first our API downloaded from the TJBOT blog required access to three conversation services:

- 1) Speech to Text,
- 2) Watson Assistant,
- 3) Text to Speech.

Listing 1. Speech to Text service authentication credentials in TJBOT's configuration file

```
exports.credentials.speech_to_text = {
  apikey: '*****',
  url:
    'https://gateway-lon.watsonplatform.net/
    speech-to-text/api/'
};
```

Basically, TJBOT was designed to co-operate with IBM Watson. However, this co-operation has a couple of limitations. One of the limitations is time restriction to use services mentioned above (minutes per month) for free. The significant factor is the number of supported languages. Besides that it is also the mentioned conflict of diode and speaker. One of the possible solutions is to modify hardware of TJBOT, build own deep neural network and use cognition services of another provider, e.g. the Google services.

II. EVALUATING QUALITY OF CHATBOT

A main feature of TJBOT is an interaction between human and machine, powered by artificial intelligence. Microphone and camera represent its input devices and LED, servomotor, speaker are its output devices. This paper is focused on speech communication, where input is human speech and outputs are primary speech and secondary text. Signalization of three states of chatbot is provided by different colors of LED (green – ready to listening, orange – processing, blue – generating answer). The input is natural language and outputs are responses generated by machine (between person and non-person). This characteristic fits into the definition of a chatbot. “Currently, chatbots are powered by rules-driven engines or artificial intelligent engines that interact with users via a text-based interface primarily [7].”

Intuitively, we could consider to use the modified “Turing test” for testing the intelligence of chatbot [8]. Many artificial intelligence researchers tend to opinion, that the test is not significant and useful for a research [9]–[11].

A couple of popular chatbots have been developed during their existence, e. g. Eliza, PARRY, CONVERSE, ALICE [12], Siri, Alexa, Google Assistant, Cortana [13], for helpdesks, business, education or just for fun (e. g. storytelling [14]). They have been based on cloud services with different kind of artificial intelligence techniques. Originally, TJBOT was developed to utilize IBM Watson services – Watson assistant, Watson speech to text and Watson text to speech as a part of Watson natural language processing.

Testing of chatbot's quality has many techniques, frameworks and metrics (user, message, bot) [15]. A chatbot can be tested as a software (e. g. black box, white box testing). It could be tested from a commercial point of view [16]. In this case, the black box approach roughly means “user

satisfaction” with obtained information [12]. Przegalinska et al. reported a testing method, which links neuroscientist methods, text mining and machine learning [17]. Also, a trust of chatbots' answers appears to be very important for a testing [17], [18]. Another kind of test is focused on a human-computer interaction, specifically for speech-based natural user interfaces, because it is specific feature of popular chatbots (assistants) mentioned above. The experiments took into account both the assistants (e. g. price, supported languages) and the users (e. g. age, gender) [13]. Radziwill and Benton observed quality attributes such as performance, humanity, affect, accessibility [19]. Interesting research topic is “realness” of chatbots. Morrissey and Kirakowski indicated four broad dimensions, on which the user might judge the naturalness of a chatbot. A chatbot should: be conscientious, display originality, display manners, be through [20].

In this paper, we use a simple black box test, where users talk to our TJBOT on any topic. They do not know what kind of text sources were used to train the TJBOT's DNN.

III. MODIFICATION OF TJBOT

We used for our experiment the TJBOT CZ Edition kit. However, we did not use the CZ Edition recipes and codes. We adapted the original solution [21].

A. Hardware changes

First of all, we did some hardware modifications towards the original version of TJBOT. The KY-016 RGB LED was replaced with the NeoPixel Diffused 8mm Through-Hole LED. Since this solution had the problem that speaker and diode could not work at the same time. Therefore, the speaker was reconnected to a MAX98357 DAC/Amplifier using the Raspberry PI's integrated I2S interface. This required moving the NeoPixel LED to an another pin and modifying TJBOT's Node.js library code. A small modification to Linux build script was also needed to allow building the MAX98357 kernel module. The required device tree overlay was already present in the OS Raspbian operating system. Because TJBOT made from a laser cut cardboard was very fragile, we downloaded the design files [21] and used 3D printer to create a plastic version (Fig. 3). The whole body consists of 14 pieces, which were printed.

B. Deep learning neural network architecture

Deep neural networks can be found in many scientific fields as well as in many daily used applications e.g. voice and image recognition, assistant services or classification. The main part of this work is neural network itself, which is the brain of TJBOT. The recurrent neural networks has been used [22] as a replacement of the original Watson Assistant service.

1) *Model*: The chat back-end uses a four layer, 113 million parameter byte-level Grid Gated Recurrent Unit (GRU) model. The embedding size is 1024, and each layer's hidden dimension is 2048. Grid GRU architecture uses Grid LSTM-like gating in the depth dimension, but replaces Long Short-Term Memory (LSTM) gates with simpler GRU gates. Drop-out with probability 0.1 was applied to the output of each



Fig. 2. 3D print of TJBOT

layer, and each time-step drop-out with probability 0.01 was applied to all update gates in time dimension.

$$\begin{aligned}
 u &= \sigma(b_u + x_t w_{xtu} + h_{t-1} w_{htu}) \\
 r &= \sigma(b_r + x_t w_{xtr} + h_{t-1} w_{htr}) \\
 hc &= \tanh(b_{hc} + x_t w_{xtc} + (r * h_{t-1}) w_{htc}) \\
 h_t &= h_{t-1} * (1 - u) + hc * u \\
 u_d &= \sigma(b_{ud} + x_t w_{xdu} + h_t w_{hdu}) \\
 r_d &= \sigma(b_{rd} + x_t w_{xdr} + h_t w_{hdr}) \\
 hc_d &= \tanh(b_{hcd} + (r_d * x_t) w_{xdc} + h_t w_{hdc}) \\
 y_t &= x_t * (1 - u_d) + hc_d * u_d.
 \end{aligned} \tag{1}$$

Where u, r, u_d, r_d are update and reset gates for time and depth dimensions, hc, hc_d are output candidates, $b_u, b_r, b_{hc}, b_{ud}, b_{rd}, b_{hcd}$ are biases, x_t, y_t is the input and output, $w_{xtu}, w_{htu}, w_{xtr}, w_{htr}, w_{xtc}, w_{htc}, w_{xdu}, w_{hdu}, w_{xdr}, w_{hdr}, w_{xdc}, w_{hdc}$ are layer weights, and h_t, h_{t-1} are hidden states for the current and previous time-steps.

2) *Dataset*: We used three training datasets (Table I, stats are shown in Table II). The first dataset (Dataset 1) and the second dataset (Dataset 2) for the model consists of primarily English-language text data collected from user interactions with the bot and interactions in the groups on Telegram, Discord and Second Life platforms, where the bot was involved. A small amount of literature is also included in the datasets, including all of Shakespeare's works and the Bible. The third dataset's (Dataset 3) corpus contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts [23].

TABLE I
USED DATASETS

Dataset	1	2	3
Dataset size	372 MB	454 MB	17 MB
Tokenizer	Byte	Subword	Byte
Avg bytes per token	1	1.259	1
Implementation	torch-rnn	pytorch-rnn	pytorch-rnn
Vocabulary size	218	264	140
Number of layers	4	4	3
Layer size	1024 × 2048	1024 × 2048	512 × 1024
Dropout	0.1	0.1	0.3
Zoneout	0.01	0.01	0.05
Epochs	16	16	20
Initial LR	3e-4	3e-4	2e-3

When using subword tokenizer, the data is initially tokenized using a byte-level vocabulary and pairs of tokens are counted. Every time the count of a pair reaches 900000, a merged token is added to the vocabulary and tokenization is restarted. Certain characters, such as space and newline, are excluded from merging.

TABLE II
STATISTICS OF THE PERSONA-CHAT DATA USED IN OUR EXPERIMENTS
FOR CHATBOT TRAINING

Attribut/ Value	Dataset 1		Dataset 2		Dataset 3	
	training	test	training	test	training	test
Number of lines	12423727	31453	15129178	40240	271932	32781
Number of sentences	13849584	35368	16765119	45811	477768	58982
Unique sentences	8441172	29579	10476735	37067	390082	51016
Unique words	965104	23040	1141607	126749	58245	18925
Words	73853556	220609	90248666	271791	3032296	376096

3) *Training*: Each training dataset was concatenated into one text file with messages separated by newlines and initially split into sequences of length 256. Initial batch size was 128. On epochs 6, 10 and 14 the sequence length was doubled and batch size was halved. Batches were shuffled every epoch, and on odd epochs the dataset was shifted by seq_length/2. Truncated backpropagation through time was used to train the network with Adam as the adaptive learning rate optimization algorithm [24]. Initial learning rate was set to 0.0003. The learning rate was halved every two epochs. The model was trained on one NVIDIA GeForce 1080 Ti video card and the training took 18 days, 9 hours and 34 minutes.

C. Switch to Google services

Communication between user and TJBOT originally ran through IBM Watson services, too – Speech to Text and Text to Speech. Their limitation in number of supported languages, which was mentioned above, was resolved by replacement with the Google's services.

1) *Google speech-to-text and vice versa*: The speech-to-text and text-to-speech functionality in the TJBOT library was replaced with Node.js libraries for Google speech-to-text [25] and text-to-speech services [26]. The keys for

an authentication credentials to IBM Watson services were replaced by keys to Google services (see Listing 2).

Listing 2. Key for Google in googleacc.json

```
{
  "type": "service_account",
  "project_id": "****-****-****",
  "private_key_id": "*****",
  "private_key": "-----BEGIN PRIVATE KEY-----\n (...)",
  "client_email": "****@****.iam.gserviceaccount.com",
  "client_id": "****",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/****.iam.gserviceaccount.com"
}
```

The key is loaded during an initialization of client (see Listing 3).

Listing 3. Key load

```
{
  const client = new speech.SpeechClient({
    projectId: '****-****-****',
    keyFile: './googleacc.json',
  });
}
```

2) *Translation integration:* As the chatbot’s neural network has been trained for English language, translation service has to be taken into account. Google’s translation service was used for translating user’s messages from user’s language to English and translating responses from English back to the user’s language.

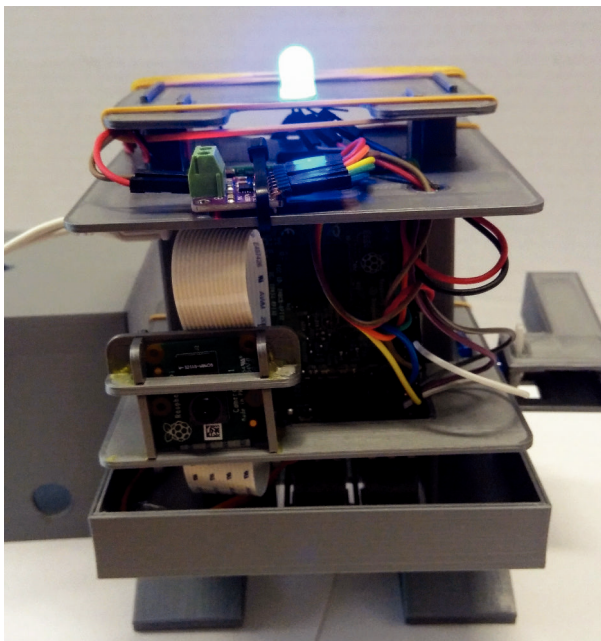


Fig. 3. Chatting TJBOT

IV. RESULTS

After some hardware modifications of the TJBOT and interconnection of the Google cloud services for speech

recognition and generation with the chatbot neural network, it can be used as a talking chatbot in user selected language. An example of chat transcription is presented in Listing 4.

Listing 4. Example of chat

```
InSK: "Ahoj"
InEN: "Hi"
OutEN: "Hello."
OutSK: "Ahoj."
InSK: "ako sa máš"
InEN: "how are u"
OutEN: "I am fine. How are you?"
OutSK: "Mám sa dobre. Ako sa máš?"
InSK: "som unavený"
InEN: "I am tired"
OutEN: "That's good."
OutSK: "To je dobré."
InSK: "pomôžeš mi napísať článok"
InEN: "help me write an article"
OutEN: "I can think of a few things"
OutSK: "Napadá ma pár vecí"
InSK: "Tak hovor"
InEN: "So talk"
OutEN: "This is a good place to speak too."
OutSK: "To je dobré miesto hovoriť taky."
InSK: "Aký máš nápad"
InEN: "What's your idea"
OutEN: "It is used to show the location of the second life site that is not being used in the future."
OutSK: "To sa používa na zobrazenie polohy druhého života webu, ktorý nie je používaný v budúcnosti."
InSK: "Nerozumím ti"
InEN: "I can not understand you"
OutEN: "It is not the best way to start the work."
OutSK: "To nie je najlepší spôsob, ako začať pracovať."
InSK: "idem preč"
InEN: "I'm going away"
OutEN: "It's okay. I am really tired."
OutSK: "Je to v poriadku. Som naozaj unavený."
InSK: "dobrú noc"
InEN: "good night"
OutEN: "Good night."
OutSK: "Dobrá noc."
```

Each four lines present one turn of conversation starting with user speaking in Slovak language. Recorded speech is converted to text as InSK and translated to English by Google translate service as InEN. This text is used as an input for chatbot returning the answer as OutEN, which is translated back to Slovak language and played back to user as OutSK. Character of training data for chatbot’s neural network is quite evident in some generated answers. Another problem arises from inaccuracy of translation between user’s language (Slovak) and chatbot language (English).

A. Testing quality of chatbot

Our testing of chatbot’s quality was done by simple evaluation of its responses in common speech dialogues between human and TJBOT as chatbot in Slovak language. Each tester played three dialogs, one for each neural network, with chatbot in ten turns and evaluates quality of the chatbot’s provided responses. Evaluation was done in a form of survey, where participants evaluated each chatbot’s response by choosing one of the following options: relevant answer, relevant answer but to different topic, relevant answer but wrong translation, inappropriate answer or not evaluated answer. Results of the black box testing are summarized in figure 4.

All participants involved in the chatbot evaluation had not talked to this chatbot before. They were asked to talk to chatbot on randomly selected topics without prior knowledge about character of training data, which were used for neural network learning. According to results, it is evident that neural network 1 trained on chats was capable of providing better

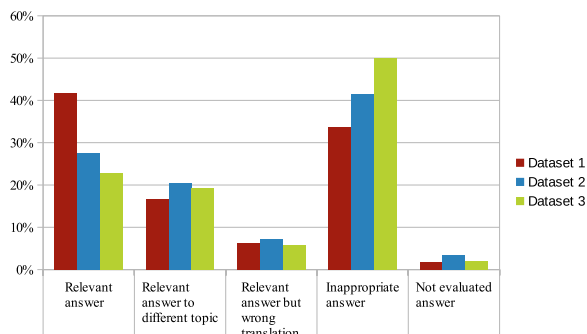


Fig. 4. Evaluation of chatbot's responses

answers (41.7 %) than the neural network 3 trained on movie scripts (22.8 %). Possible reason is that conversations during the testing chatbot were more similar to conversations used in chats as training data. Reason for low relevance of responses of the neural network 2 could be due to selection of inappropriate sample for training data of the neural network. In all test, approximately 18 % of answers were marked as relevant to another topic, which means that chatbot did not identify conversation topic correctly.

Another issue is the language translation problem. Chatbot design allows to have discussion not only in English language, which were used in neural network training, but also in other languages, which are supported by the Google translate services. Quality of translation has also impact on the quality of chatbot. Regardless of neural network used in test more than 6 % of answers were marked as relevant, but translation significantly shifts the meaning of the replied answers. This problem can be diminished by improving Google translate service accuracy.

V. CONCLUSION

We have successfully adapt the IBM TjBot as an interface for chatbot with the Google cloud services involved. Cloud services like speech-to-text, text-to-speech and translate service has been used as an interface to the chatbot conversation. Chatbot answering logic was implemented by trained deep neural network.

In the future, we plan to do some more improvements. From hardware point of view rechargeable Li-ion based power source charged from USB power source will be added to allow TjBot to operate fully wirelessly. Form software point of view quality of answers could be further increased by training more sophisticated chatbot's deep neural network to be capable of answering questions in specific domain such as questions about faculty and studies. This would allow as to study more aspects of conversation, not only the relevance of the replied answers.

ACKNOWLEDGMENT

This work was supported by IT Academy – Education for the 21st Century ITMS 312011F057 and the Slovak Grant

Agency KEGA contract No. 011UMB-4/2017. This research was also supported by IBM. We thanks IBM for granting access to IBM Cloud services and IBM Watson services.

REFERENCES

- [1] lbmtjbot.github.io. (n.d.). "IBM TjBot." [online] Available at: <https://lbmtjbot.github.io/> [Accessed 2 Sep. 2019].
- [2] M. R. Ebling. 2017. "TjBot and Zooids: The Connection between Pervasive Computing and AI." in IEEE Pervasive Computing, vol. 16, no. 1, pp. 2–4, doi: 10.1109/MPRV.2017.
- [3] V. C. Dibia, M. Ashoori, A. Cox, and J. D. Weisz. 2017. "TjBot: An Open Source DIY Cardboard Robot for Programming Cognitive Systems." in Proceedings of the CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17). ACM, New York, NY, USA, 2017, pp. 381–384. DOI: <https://doi.org/10.1145/3027063.3052965>
- [4] Tjbotcz.github.io. (n.d.). "TjBot – Roztomil robot z laborato IBM aneb uml inteligence v kartonu." [online] Available at: <https://tjbotcz.github.io/> [Accessed 2 Sep. 2019].
- [5] GitHub. (n.d.). "TjBot tjbotcz." [online] Available at: <https://github.com/tjbotcz> [Accessed 4 Sep. 2019].
- [6] Raspberrypi.org. (n.d.). "Which pin(s) on RPi 3B is PWM capable – Raspberry Pi Forums." [online] Available at: <https://www.raspberrypi.org/forums/viewtopic.php?p=1425176> [Accessed 8 Sep. 2019].
- [7] R. Khan, D. Anik. 2018. "Introduction to chatbots." Build Better Chatbots. Apress, Berkeley, CA, pp. 1–11.
- [8] A. M. Turing. 1950. "I.COMPUTING MACHINERY AND INTELLIGENCE." Mind, Volume LIX, Issue 236, October 1950, Pages 433-460, <https://doi.org/10.1093/mind/LIX.236.433>
- [9] A. Sloman. 2014. "Judging chatbots at turing test 2014." [online] Available at: <https://www.cs.bham.ac.uk/research/projects/cogaff/misc/turing-test-2014.pdf> [Accessed 15 Oct. 2019].
- [10] J. H. Moor. 2001. "The status and future of the Turing test." Minds and Machines, 11.1: pp. 77–93.
- [11] R. French. 2012. "Moving Beyond the Turing Test." Communications of the ACM. 55. 74–77. 10.1145/2380656.2380674.
- [12] B. A. Shawar, E. Atwell. 2007. "Different measurements metrics to evaluate a chatbot system." In: Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies. Association for Computational Linguistics, 2007. p. 89–96.
- [13] G. Lpez, L. Quesada, L. A. Guerrero. 2017. "Alexa vs. Siri vs. Cortana vs. Google Assistant: a comparison of speech-based natural user interfaces." In: International Conference on Applied Human Factors and Ergonomics. Springer, Cham, p. 241–250.
- [14] C. Curry, J. D. O'Shea. 2012. "The implementation of a story telling chatbot." Adv.Smart Syst.Res., 1(1), pp. 45.
- [15] AI Multiple. 2019. "15 Key Metrics for Chatbot Conversational Analytics in 2019." [online] Available at: <https://blog.aimultiple.com/chatbot-analytics/> [Accessed 11 Sep. 2019].
- [16] J. Dunn. 2016. "We put Siri, Alexa, Google Assistant, and Cortana through a marathon of tests to see who's winning the virtual assistant race: here's what we found." Business Insider. [online] Available at: <https://cloud.google.com/text-to-speech/> [Accessed 15 Oct. 2019].
- [17] A. Przegalinska, L. Ciechanowski, et al. 2019. "In bot we trust: A new methodology of chatbot performance measures." Business Horizons, ISSN 0007–6813, <https://doi.org/10.1016/j.bushor.2019.08.005>. [online] Available at: <http://www.sciencedirect.com/science/article/pii/S000768131930117X> [Accessed 15 Oct. 2019].
- [18] C. B. Nordheim, A. Følstad, et al. 2019. "An Initial Model of Trust in Chatbots for Customer Service—Findings from a Questionnaire Study." Interacting with Computers. Oxford University Press, iwz022, <https://doi.org/10.1093/iwc/iwz022>
- [19] N. M. Radziwill, M. C. Benton. 2017. "Evaluating quality of chatbots and intelligent conversational agents." arXiv preprint arXiv:1704.04579.
- [20] K. Morrissey, J. Kirakowski. 2013. "Realness 'in Chatbots: Establishing Quantifiable Criteria." In: International Conference on Human-Computer Interaction. Springer, Berlin, Heidelberg. p. 87–96.

- [21] M. Ashoori, (n.d.). "TJBOT - Build a Talking Robot." [online] Instructables circuits. Available at: <https://www.instructables.com/id/Build-a-Talking-Robot-With-Watson-and-Raspberry-Pi/> [Accessed 9 Sep. 2019].
- [22] A. Karpathy. 2015. "The Unreasonable Effectiveness of Recurrent Neural Networks." [online] Karpathy.github.io. Available at: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/> [Accessed 15 Oct. 2019].
- [23] C. Danescu-Niculescu-Mizil, L. Lee. 2011. "Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs." In: Proceedings of the 2nd workshop on cognitive modeling and computational linguistics. Association for Computational Linguistics. p. 76–87.
- [24] D. P. Kingma, J. L. Ba. 2014. "Adam : A method for stochastic optimization." arXiv:1412.6980v9
- [25] Google Cloud. (n.d.). "Cloud Speech-to-Text - Speech Recognition." [online] Available at: <https://cloud.google.com/speech-to-text/> [Accessed 8 Sep. 2019].
- [26] Google Cloud. (n.d.). "Cloud Text-to-Speech - Speech Synthesis." [online] Available at: <https://cloud.google.com/text-to-speech/> [Accessed 8 Sep. 2019].

An Approach to Web Advertising Server Parsing - Apartment Price Analysis in the Czech republic

Alena Pozdilková

University of Pardubice
Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
alena.pozdilkova@upce.cz

Marie Nedvědová

University of Pardubice
Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
st36806@student.upce.cz

Jaroslav Marek

University of Pardubice
Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
jaroslav.marek@upce.cz

Abstract— The real estate market is not subject to frequent analyzes in the academic field, which is due to the difficulty in obtaining data. By automatic polling, we are able to get data on the floor area of advertised apartments and the asked purchase price. A Python script was written to retrieve data from srealty.cz. At 2:30 am every day, the script downloaded ads from the website srealty.cz. The MongoDB database is used to store ads. New ads are saved directly to the database. If an ad has already been stored in the database, it will only be updated if needed (price change, etc.). Then, daily and monthly summaries for used filters are created in the database. The filtered data can then be displayed or exported to a file via the web interface. We will present analysis of data in various municipalities of the Czech Republic in the period 09/2018 – 08/2019.

Keywords—real estate market; time series; apartment prices

I. INTRODUCTION

In this article, we will analyze apartment prices in the Czech Republic in the period starting in July 2018. We will not have data from real estate transfers, but we will only work with data from advertisements. Occasionally, data provided by large real estate companies appear in the media. But this information may be affected by the interests of real estate brokers and may not best map the real estate market behavior. We can also see predictions of property price developments created by economic analysts. In our study, we will process the data we collect through long-term collection from large relative servers. By parsing web pages, you can get prices from “apartments for sale” ads. The source of our data are ads of srealty.cz.

The srealty.cz website offers an open web (HTTP) API that allows you to read and manipulate advertisements on the server. Only the reading part was used for the purposes of statistical evaluation. Downloading ads uses HTTP GET request and results are passed in JSON format. Authentication is not required to access ads because the data is publicly available.

By simple calculation we get the converted price for one square meter. Similarly, prices for houses or land could also be monitored. However here, the averaging (more floors, incompatibility of reported areas: floor x built-up) is much more complicated and therefore we will only deal with flats. Our goal will not be to predict future price developments, but only to compare apartment price characteristics in different locations. Cf. (Risse [6]). A guide to our analysis was provided by the book (Winson-Geideman [8]). Real estate data is often taken from LAG models, see (Guo [2]) and (Pozdilková [5]). The issue of investing in real estate is also examined from the perspective of price bubble prediction. There are studies examining the risk of investing in real estate and comparing profitability with the stock exchange, cf. (Szumilo [7]). However, we will not perform spatial analysis or profitability analysis in this article.

II. PARSING

In this section we describe the programmed application for retrieving data from real estate servers.

A. Software solution

The data reserved for our research come from srealty.cz. This is some kind of web pages containing ads. Each advert was recorded separately using a Python script. Cf. (Oliphant [4]) and (McKinney [3]). At 2:30 am every day, the script tries to download all ads from each district from the website srealty.cz. The MongoDB database is used to store ads. New ads are saved directly to the database. If an ad has already been stored in the database, it will only be updated if needed (price change, etc.). Then, daily and monthly summaries for all used filters are created in the database. The filtered data can then be displayed or exported to a file after access via the web interface. Finally, the data are ready to be examined.

B. Architecture of data mining solution

The complete data mining solution is built using the Docker container virtualization platform. A total of 3 containers were used:

- (a) applications that download and update data.,
- (b) web application for presentation of results,
- (c) MongoDB database server used for data persistence.

MongoDB database was chosen for persistence of data of individual advertisements. This NoSQL database provides not only means for persistence of data. Complex aggregation pipelines are widely used in the processing of downloaded data. The pipelines perform transformation, grouping and resulting calculation of results. The auxiliary aggregate values are further stored in the "cache" collection in the database. In the future, it is possible to count on the use of sharing for managing larger volumes of recorded data.

Both the download and result presentation applications are programmed in Python (version 3.7) and run as separate containers. The application for downloading data uses the Python core library and the pymongo library to communicate with the database. The web application is built over the Flask library and also uses the pymongo, pandas and pillow libraries.

The Docker Container Platform allows the entire solution to be easily run in the target environment and offers the option of an easy transition to the Docker Swarm platform if performance is not sufficient in the future. Both the database and the web application can be scaled horizontally to improve performance. In (Cook [7]) are some examples of Docker usage.

C. Download data

Application for downloading data is basically quite simple script that optimally every day at 2:30 performs all operations. If problems occur, the download will be retried later. Data loading itself is a trivial use of the Sreality API. All districts and all advertisements are scanned. For individual advertisements, it is first tested if it is already present in the database - a combination of hash id, ad title and type of advertisement is used for verification. If the advertisement is already present, the existing record will be updated and current values will be added. If the advertisement is not in the database at all, the loaded advertisement is transformed into the required form and saved in the database.

After all advertisements are downloaded, aggregate pipelines are started in the database. In the first phase, daily summaries are created by municipality and district. In the second phase of the algorithm, monthly aggregations are created in a similar way. After the aggregations are completed, the results can be presented using a web application.

D. The web application

The application is built on a simple Flask framework. Several basic functions were included in the application during programming:

- export of selected statistics with the possibility to select and apply filtering (for example, apartments 1 + 1, 1 + 2, 1 + 3),
- display a map of all ads above the real map background,
- display a map of districts of the Czech Republic with average values of prices per square meter.

The basic function is to transform data from the database and present them in CSV format. The necessary data are read directly from the database and using the pandas' library, organized into the resulting table and exported to the client.

To verify the functionality of the whole solution, the support was created for displaying the map of the Czech Republic with rendering of individual ads. The Open Layers library and a map file from the OpenStreetMap service were used for the implementation. Advertisements are passed as a separate map layer above the default map background. Rendering is done using the python and pillow library.

The last functionality of the application is displaying a map of districts of the Czech Republic and information on the total price for individual months with the possibility to compare and evaluate the development of prices over time.

III. EXPLANATORY ANALYSIS

Through the parsing of the real estate website, we have acquired data for the past year containing the municipality identifier, the apartment price, the apartment area and the date of the advertisement. The positions of all sites with ads are rendered in the Fig. 1. Examples of such data are given in Tab. 1. On 30th April 2019 the total number of ads analyzed was 16748. Overall, in the period between September 1st, 2018 and August 30th, 2019, we loaded 6,845,000 ads in 250 working days. The location of the sites with ads is shown on the map. The GPS coordinates of each settlement were obtained during parsing.

A. Basic Analysis

For each municipality that appeared in at least one ad, we calculated the average value and the standard deviation of the price of a one square meter of apartment. The example of measured values for chosen region is given in Table I.

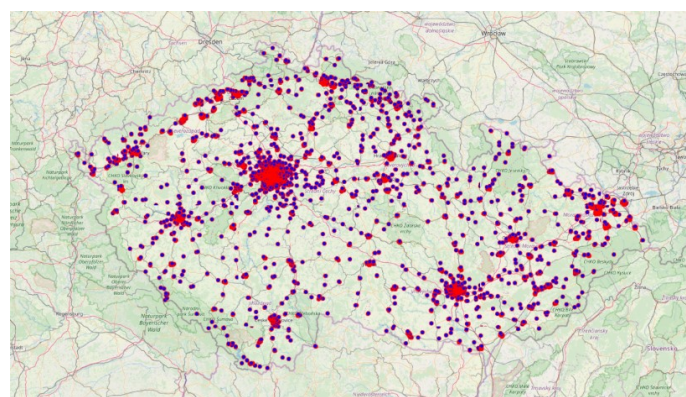


Fig. 1. Distribution of advertisement. Source: own.

TABLE I. SOURCE DATA

Data		Date	Price	Area	Prize of 1 m ²
City	Žďár nad Sázavou	2019-08-05	2400000 Kč	74.0	324332.4 Kč
		2019-08-05	2400000 Kč	57.0	29122.8
		2019-08-05	2400000 Kč	63.0	28412.8
		2019-08-05	2400000 Kč	68.0	30882.4
		2019-08-05	2400000 Kč	52.0	26153.8
		2019-08-05	2400000 Kč	123.0	32113.8
		2019-08-05	2400000 Kč	71.0	25338.0
		2019-08-05	2400000 Kč	78.0	33961.54

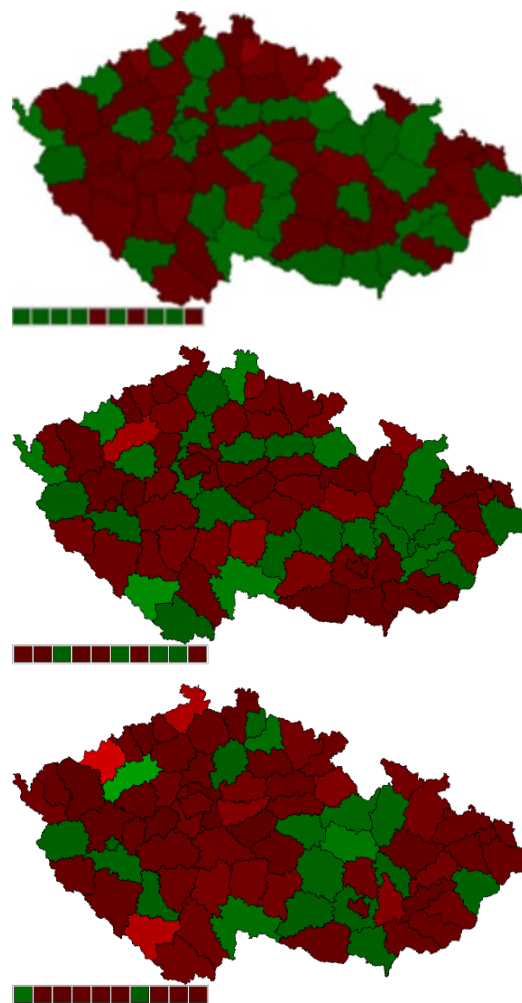
a. Source: own.

TABLE II. AN EXAMPLE OF TEXT FEATURE VALUES

City	No. of Ads	Average price of 1 m ²	Standard deviation
Žďár nad Sázavou	8	29802.2	2880.1
Velká Bíteš	2	35682.5	341.6
Nové Město na Moravě	3	49401.8	3704.6
Lavičky	5	40219.9	502.9
Rozsochy	1	15000.0	
Velké Meziříčí	1	29649.1	
Bystřice nad Pernštejnem	3	32398.8	3312.9
Osová Bitýška	1	24105.3	

b. Source: own.

From the obtained data in the structure of Tab. 1, we can get basic statistical characteristics (number of ads, average daily price in a given municipality, standard deviation in a given municipality). An example of these variables is given in Tab. 2. These values can be accumulated over a longer period. This gives us time series of prices for all municipalities with advertised apartments. This data can be studied using basic methods for time series processing. Time series in the reference period allow to assess how the change in the rules for mortgage lending in the autumn of 2018 affected average prices. In Figure 2 at the top left, shows that between October 31st, 2018 and April 30th, 2019 the overall average price in about half of the districts decreased, despite an increase in the middle of this interval. In the figure, the districts where prices have risen, are shown in red. Districts with decreasing price per 1 m² of flat are depicted in green. The stripe of the colored squares shows the progression of prices in the districts of Prague 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. The second map at the top right demonstrates what price changes occurred between October 2018 and January 2019. The only difference is in a small number of districts, for example in South Moravia. In the first quarter of 2019, the number of red districts is increasing and the upward trend in prices is beginning to emerge. Of course, price fluctuations, see Figure 5 on the bottom left, can of course be influenced by only a few developers' projects, which will increase the average price, especially in small districts.

Fig. 2. Histogram of average prices for 1m² for all ads. April 2019 versus October 2018. January 2019 versus October 2018. April 2019 versus January 2019. Source: own.

B. Price development

With the help of basic graphic tools, we will present to the reader some interesting facts. Figure 3 and 4 shows the histograms of average prices from 1st September 2018 to 30th August 2019 in two selected districts. It can be seen that the small district of Žďár nad Sázavou has a greater kurtosis and a smaller variability than Prague 10. This effect was also apparent in other districts.

Of course, price developments can best be grasped by showing time series. Figure 5 - 8 shows the graph of price development (in thousands of CZK) and the number of advertisements in Prague 10 and Žďár nad Sázavou. It shows how the number of ads is growing. In the middle of the horizontal axis is November 2018, when conditions for providing mortgages were tightened. The charts show growth in the number of ads and also the growth in prices before changes. The average price then falls in Prague only for a short period of time, when the price increase is clearly related to the decline in supply. In Žďár nad Sázavou, the declining price correction remains for the first quarter of 2019.

In the period under review, the average price calculated from 6,845,000 ads was 55,075 CZK.

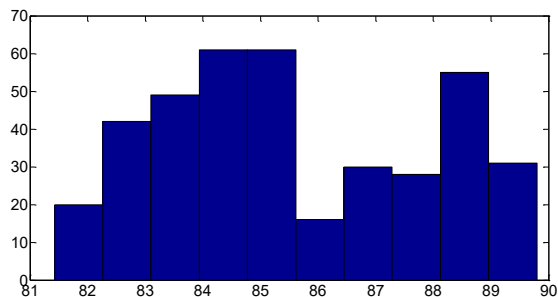


Fig. 3. Prague 10: histogram of price per square meter [thousand CZK/m²] in considered period. Source: own.

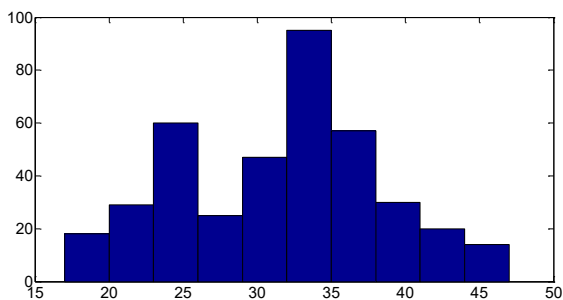


Fig. 4. Žďár nad Sázavou: histogram of price per square meter [thousand CZK/m²] in considered period. Source: own.

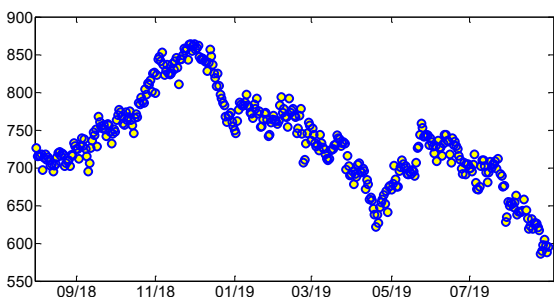


Fig. 5. The number of ads – region Prague 10. Source: own.

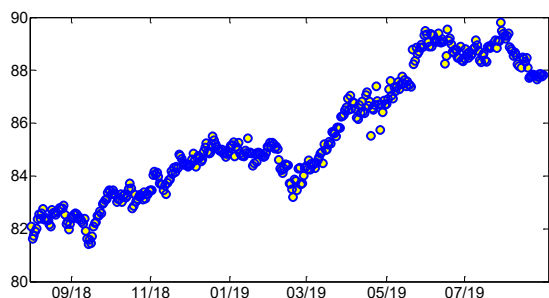


Fig. 6. Price graph [thousand CZK] – region Prague 10. Source: own.

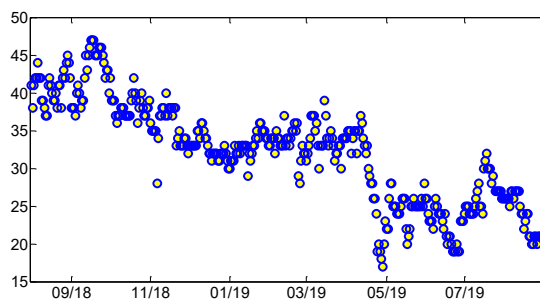


Fig. 7. The number of ads – Žďár nad Sázavou (supply decrease). Source: own.

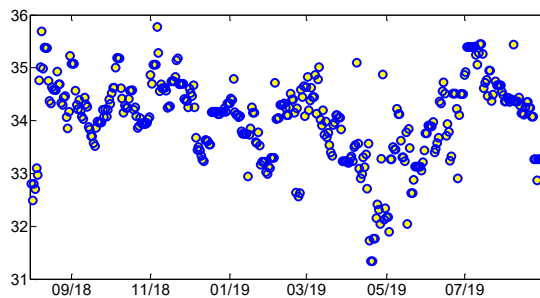


Fig. 8. Price graph – Žďár nad Sázavou. Source: own.

IV. CONCLUDING REMARKS

The gist of article is a description of a software solution for collecting information about advertised apartments from real estate servers. The designed and implemented platform enables automated data collection from the Sreality service and their automatic processing within the long-term operation. Platform functionality can also be expanded in the future and, if necessary, vertical and horizontal scaling can be used to increase platform performance. The obtained data leads to the opportunity of further research in this field of study.

ACKNOWLEDGMENT

This research was supported by the Internal Grant Agency of University of Pardubice, the project SGS 2019 024.

REFERENCES

- [1] J. Cook, “Docker”. In: Docker for Data Science. Apress, Berkeley, CA, 2017.
- [2] J. Guo, and. X. Qu, “Spatial interactive effects on housing prices in Shanghai and Beijing.” Regional Science and Urban Economics, 2018, no. 1, pp. 1–14.
- [3] W. McKinney, “Data structures for statistical computing in python.” In: Proceedings of the 9th Python in Science Conference. June 2010, vol. 445, pp. 51-56.
- [4] T. E., Oliphant, “Python for scientific computing.” In: Computing in Science & Engineering. May-June 2007, vol. 9, no. 3, pp. 10-20.
- [5] A. Pozdílková, and J. Marek, “Spatial lag model for apartment prices in Pardubice region.” In: D. Szarkova, D. Richtarikova, P. Letavaj, J. Gabkova (Eds.), Proceedings of 17th Conference on Applied Mathematics Aplimat 2018, Bratislava: Slovak University of Technology Bratislava, February 2018, pp. 867–875.

- [6] M. Risse, and. M. Kern, "Forecasting house-price growth in the Euro area with dynamic model averaging." *North American Journal of Economics and Finance*, vol. 38, 2016, pp. 70–85.
- [7] N. Szumilo, T. Wiegelmann, E. Łaskiewicz, M. Pietrzak, M. Bernard, A.P. Balcerzak, "The real alternative? A comparison of German real estate returns with bonds and stocks." *Journal of Property Investment & Finance*, 2018, vol. 36, no. 1, pp. 19-31.
- [8] Winson-Geideman, K., Krause, A., Lipscomb, C. A., Evangelopoulos, N. "Real Estate Analysis in the Information Age: Techniques for Big Data and Statistical modelling." Routledge, Abingdon: Oxon. 2017.

Totally bounded metric spaces and similarity detecting algorithms

Gábor Sági

Alfréd Rényi Institute of Mathematics,
Reáltanoda u. 13-15,
H-1053 Budapest, Hungary
and
Budapest University of
Technology and Economics,
Department of Algebra,
Egry J. u. 1,
H-1111 Budapest, Hungary
Email: sagi@renyi.hu

Karrar Al-Sabti

Budapest University of
Technology and Economics,
Department of Algebra,
Egry J. u. 1,
H-1111 Budapest, Hungary
Email: karrar.al-sabti@edu.bme.hu
and
University of Kufa,
Faculty of Computer Science and Mathematics,
Email: karrard.alsabti@uokufa.edu.iq

Abstract—Many questions of theoretical computer science can be reduced to the following problem: let $\mathcal{X} = \langle X, \varrho \rangle$ be a metric space, let $A \subseteq X$ and let ε be a positive real number; for a given input $x \in X$ find $a \in A$ (if any) for which $\varrho(a, x) \leq \varepsilon$. This problem is called the similarity detecting problem of $(\mathcal{X}, A, \varepsilon)$. Usually, A (or sometimes X) is finite but huge, and the challenge is to represent the metric space in such a way computer algorithms may handle it efficiently.

Based on recent results of [9] we propose a similarity detecting algorithm. We associate a finite dimensional Euclidean space \mathcal{Y} to \mathcal{X} and an “almost isometry” $f : X \rightarrow Y$ which preserve distances modulo a controlled amount of inaccuracy. After that, instead of working with \mathcal{X} , we can work with \mathcal{Y} . The main result of this work is the description of the above method.

In the special case, when \mathcal{X} itself is a large dimensional Euclidean space (with its usual Euclidean metric), our method can be considered as a kind of dimension reduction. In this special case we are analyzing the time complexity of our proposed algorithm, as well.

I. INTRODUCTION

The present work has a practical motivation based on a recent more theoretical result of [9]. We start by the practical motivation.

Many questions of theoretical computer science can be reduced to questions about certain metric spaces, for further details we refer to [2], [6] and the references therein. Usually, these spaces are finite, but huge and the problem is how

to handle these spaces by computer algorithms in a tractable way. This is the case, for example, if the distance function of the metric space measures “similarity” of two objects and the problem is to find the elements of a database which are similar enough to a given input. Related problems can be effectively solved if one is able to represent the metric space in a suitable way, for example, if one is able to embed the metric space into a finite dimensional Euclidean space (endowed with its usual metric) with a function which is an “almost isometry”, or if one is able to embed a compact subset of a (finite dimensional) Euclidean space having large dimension into a considerably smaller dimensional Euclidean space by an “almost isometry”.

For the theoretical background we briefly recall investigations initiated in [10] and continued in [9]. For metric spaces $\mathcal{X} = \langle X, \varrho \rangle$, $\mathcal{Y} = \langle Y, \sigma \rangle$ and a positive real number ε , a function $f : X \rightarrow Y$ is defined to be an ε -map iff for all $y \in Y$, the diameter of $f^{-1}(y)$ is at most ε . Thus, if ε is small, then f is almost injective. In Theorem 10 of [9] the first author gave a new proof for the following well known fact: if \mathcal{X} is totally bounded (for further explanation see Definition 2.2 below), then for all ε there exists a finite number n and a continuous ε -map $f_\varepsilon : X \rightarrow \mathbb{R}^n$, where \mathbb{R}^n is the usual n -dimensional Euclidean space endowed with the Euclidean metric. Such ε -maps still exist even if \mathcal{X} has infinite covering dimension (in this case, n depends on ε , of course). Contrary to the previously known proofs (see e.g. Chapter 8 of [8]), the proof technique in [9] is effective in the

sense, that it allows to establish estimations for n in terms of ε and structural properties of \mathcal{X} .

Utilizing this theoretical background, in the present work we are proposing a method which can be used for similarity detecting, clustering and related problems. The sstructure of this paper is rather simple. At the end of this section we are summing up our system of notation. In Section II we describe and analyze our similarity detecting algorithm with special emphasis for the case, when \mathcal{X} itself is a large dimensional Euclidean space. Finally in Section III we provide some conclusion.

Notation

Our notation is mostly standard, but the following list may help.

Throughout \mathbb{N} denotes the set of natural numbers. In addition, \mathbb{R} and \mathbb{R}^+ denotes the set of real numbers, and the set of positive real numbers, respectively.

Let $\mathcal{X} = \langle X, \varrho \rangle$ be a metric space, $a \in X$ and let γ be a non-negative real number. As usual, the open γ -ball $B(\gamma, a)$ at a is the set

$$B(\gamma, a) = \{x \in X : \varrho(a, x) < \gamma\}.$$

If \mathcal{E} is an Euclidean space, then the norm of any element x of \mathcal{E} will be denoted by $\|x\|$ (the norm function of \mathcal{E} and its usual metric are mutually definable from each other in the usual way).

II. A SIMILARITY DETECTING ALGORITHM

Let $\mathcal{X} = \langle X, \varrho \rangle$ be a metric space and let $A \subseteq X$ be a given set. Intuitively, X is the set of all instances of an abstract data type and ϱ measures similarity between the elements of X : if $\varrho(x, y)$ is "small" for some $x, y \in X$ then we say, that x and y are "similar enough" to each other. More concretely, we fix $\varepsilon \in \mathbb{R}^+$ and consider it as an amount of inaccuracy one can tolerate. Then "x and y are similar enough" means $\varrho(x, y) < \varepsilon$.

More formally, the similarity detecting problem for $(\mathcal{X}, A, \varepsilon)$ is the following: given an input $x \in X$ find $a \in A$ such that $\varrho(x, a) < \varepsilon$. The problem is, that A may be huge and computing ϱ for two particular points may be slow.

As we mentioned, our goal in this section is to propose and analyze an algorithm that can be used to handle the above problem efficiently. To do so, we start by recalling the theoretical background.

Definition 2.1: A family $\{B_i : i \in I\}$ of γ -balls is defined to be a γ -net iff it covers X , that

is,

$$X = \bigcup_{i \in I} B_i.$$

We recall e.g. from [1] the following definition.

Definition 2.2: A metric space \mathcal{X} is defined to be totally bounded iff for all positive $\gamma \in \mathbb{R}$ there exists a finite γ -net in \mathcal{X} .

As it is well known, \mathcal{X} is compact iff it's metric is totally bounded and complete (i.e. every Cauchy sequence is convergent in \mathcal{X}). For further details we refer to [1], as well. It is well known, that every finite metric space is compact.

If \mathcal{X} is a totally bounded metric space, then $\nu(\mathcal{X}, \gamma)$ denotes the smallest cardinality κ for which there exists a κ -sized γ -net of \mathcal{X} .

For a given $\varepsilon \in \mathbb{R}^+$ let

$$N := 6 \cdot \nu(\mathcal{X}, \frac{\varepsilon}{30}) \cdot \nu(\mathcal{X}, \frac{\varepsilon}{12}).$$

Suppose \mathcal{X} is a (finite or infinite) totally bounded metric space. Then, according to Theorem 10 of [9],

(1) if \mathcal{X} does not contain isolated points, then there exist $n \leq N$ and an ε -map $f : X \rightarrow \mathbb{R}^n$ such that, for all $x, y \in X$ we have

$$\|f(x) - f(y)\| \leq \sqrt{n}\varrho(x, y),$$

in particular, f is continuous.

(2) if \mathcal{X} has countably many isolated points, then there exist $n \leq 1 + N$ and a continuous ε -map $f : X \rightarrow \mathbb{R}^n$.

(3) if \mathcal{X} is compact, then there exist $n \leq 1 + N$ and a continuous ε -map $f : X \rightarrow \mathbb{R}^n$,

further the ε -map is effectively constructed in all cases above.

Now we can sketch our similarity detecting algorithm as follows (immediately after sketching the algorithm, we comment and further explain its crucial steps).

(Step 1) For a given $\varepsilon' \in \mathbb{R}^+$ find $n \in \mathbb{N}$ and a continuous ε' -map $f : X \rightarrow \mathbb{R}^n$;

(Step 2) compute $B := \{\langle f(a), a \rangle : a \in A\}$;

(Step 3) for an input $x \in X$ find $b =: \langle u, v \rangle \in B$ for which the usual Euclidean distance $\|f(x) - u\|$ is minimal;

(Step 4) if (for $b = \langle u, v \rangle$ computed in (Step 3) above) we have

$$\|f(x) - u\| < \varepsilon',$$

then the output is v ; otherwise there are no elements of A which are similar enough to x .

In (Step 1) above, n and f can be constructed in an algorithmic way, because in the proof of Theorem 10 of [9], the ε -maps have been constructed effectively for all ε . For (Step 2) we note, that

$$\{x : \exists y \langle x, y \rangle \in B\} = \{f(a) : a \in A\} \subseteq \mathbb{R}^n.$$

So, in (Step 3), instead of working with the original distance ϱ we are working with Euclidean distances that can be computed relatively quickly, provided that n is small enough.

A particularly important special case of the general similarity detecting problem is, when X is a subset of a large dimensional Euclidean space (and ϱ is the corresponding Euclidean distance), that is, if $X \subseteq \mathbb{R}^k$ for a large k . This special case will be called *dimension reduction*. We make the following remarks:

- Note, that (Step 1) and (Step 2) in the above sketch are preparatory: they should be performed only once at the beginning; if we are searching similar objects many times, then the cost of (Step 3) will dominate.

- If the database A changes in time, then according to (Step 2) we can quickly modify B , as well.

Now we turn to study the dimension reduction problem, that is, the similarity detecting problem, when A is a subspace of a large dimensional Euclidean space $\mathcal{X} = \mathbb{R}^k$. Of course, the critical point in our algorithm is (Step 1) in which we have to find n and f . According to Theorem 10 of [9] they exist and f can be effectively constructed from n and from given $(\frac{\varepsilon}{30})$ -nets and $(\frac{\varepsilon}{12})$ -nets of A . We have to choose n as small as we can in order to make our algorithm more efficient. The rest of the present section is devoted to investigate the choice of n for dimension reduction.

Analyzing the proof of Theorem 10 of [9], one concludes, that n becomes smaller whenever one is able to find smaller $(\frac{\varepsilon}{30})$ -nets and $(\frac{\varepsilon}{12})$ -nets of A . More generally, for given $\delta \in \mathbb{R}^+$ and $k \in \mathbb{N}$ one has to find a k -sized δ -net of A (if such exists).

The natural approach would be, that for a given δ one tries to find δ -nets of A with as small cardinalities k as possible. However, this do not would be an efficient method, as such an approach would be equivalent to solve *NP*-hard problems of computational geometry, and cluster analysis; for further details in that direction we refer to [7] and the references therein. Instead, we propose to fix k and estimate δ for which there exists a k -sized δ -net in A ; then consecutively increasing k we will obtain a decreasing sequence of the corresponding $\delta = \delta_k$ and we increase k until δ_k will be sufficiently small.

For a fixed k and a given finite $A \subseteq \mathbb{R}^m$ the well known k -center problem is to find a k -element subset $B \subseteq A$ such that

$$\max_a \min_b \{\|b - a\| : a \in A\}$$

is as small as possible (for further details we refer e.g. to [5]). This problem is related to cluster analysis and to the inverse shortest path problem. According to [3], the k -center problem (already in the Euclidean plane \mathbb{R}^2) is known to be NP-complete as well (for more recent related investigations we refer to [11] and [12]). Hence, instead of exact solutions, it is more practical to search for suboptimal, approximate solutions. Indeed, there is a classical approximate solution for the k -center problem due to Gonzalez [4]; we will briefly recall this.

For a metric space $\mathcal{X} = \langle X, \varrho \rangle$ and $A \subseteq X, b \in X$ the standard definition of the distance of A and b is

$$\varrho(A, b) = \inf \{\varrho(a, b) : a \in A\}.$$

Then the farthest path transversal sequence of a finite metric space is defined as follows: $a_0 \in X$ is arbitrary, and if a_j has already been defined for all $j < i$, then a_i is a point x in X for which

$$\varrho(\{a_j : j < i\}, x)$$

is maximal. Now we recall Gonzalez's idea presented in [4]: fix $k \in \mathbb{N}$ and let

$$r = \varrho(\{a_j : j < k\}, a_k).$$

Observe, that

$$(i) \text{ for any } i \neq j < k \text{ we have } \varrho(a_i, a_j) \geq r$$

and

$$(ii) \text{ for any } x \in X \text{ there exists } i < k \text{ such that } \varrho(a_i, x) \leq r.$$

By (ii), $\{B(a_i, r) : i < k\}$ is an r -net of size k . Further, Suppose $\{B(b_i, r') : i < k\}$ is another r' -net. Then, by the Pigeon-Hole principle, there would exist $i \neq j \leq k$ and $l < k$ such that $a_i, a_j \in B(b_l, r')$, therefore

$$r \leq \varrho(a_i, a_j) \leq 2 \cdot r'.$$

In another words, the minimal radius r' of a k -sized r' -net is at least $\frac{r}{2}$. Therefore r constructed above, is a 2-approximation of the minimal radius of a k -sized net.

Based on the above observations, our dimension reductor algorithm is built up from an initialization part and from a searching part; these may be summarized as follows (as before,

comments and explanations will be provided immediately after describing these algorithms).

Initializing part.

Input: a finite set $A \subseteq \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$.

1. Choose an arbitrary $a_0 \in A$ and let $k = 1, \varepsilon' = \infty$.
2. While $\varepsilon' > \varepsilon$ and $k < n$ Do
3. Let $r = \max_{x \in A} \varrho(\{a_j : j < k\}, x)$.
4. Let $a_k \in A$ be such that $r = \varrho(\{a_j : j < k\}, a_k)$.
5. Let $\varepsilon' := 2r$.
6. Let $k = k + 1$.
7. End(Do).
8. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ be the function that maps each $x \in \mathbb{R}^n$ onto

$$f(x) := \langle \varrho(x, a_0), \dots, \varrho(x, a_{k-1}) \rangle.$$

9. Compute a list enumerating

$$B = \{ \langle f(a), a \rangle : a \in A \}.$$

In order to keep notation simpler, we will denote the list enumerating B by B , as well.

Searching part.

Input: $x \in \mathbb{R}^n$.

1. let $m = 1$;
2. While $m \leq \text{length}(B)$ Do
3. let $\langle u, v \rangle$ be the m^{th} element of B ;
4. let $d := \|f(x) - u\|$;
5. if $d \leq \varepsilon/2$ then v is an output End(if);
6. if $d > \sqrt{k}\varepsilon$ then v is not an output End(if);
7. if $\varepsilon/2 < d \leq \sqrt{k}\varepsilon$ then
8. if $\|x - v\| \leq \varepsilon$ then v is an output End(if);
9. if $\|x - v\| > \varepsilon$ then v is not an output End(if);
10. End(if);
11. let $m := m + 1$;
12. End(Do).

We conclude this section by making some remarks on the Initializing and on the Searching parts.

Remarks on the Initializing part. Suppose the algorithm has already computed $\{a_j : j < k\}$ for some k . Then, according to step 3, $\{B(r, a_j) : j < k\}$ is an r -net. Let $f_k : \mathbb{R}^n \rightarrow \mathbb{R}^k$ be the function that maps each $x \in \mathbb{R}^n$ onto

$$f_k(x) := \langle \varrho(x, a_0), \dots, \varrho(x, a_{k-1}) \rangle$$

(so, according to step 8, f is f_k for the last (largest) value of k). It follows from Lemma 1 of [9], that if $x, y \in A$ are such that $\varrho(x, y) > 2r$ then $\|f_k(x) - f_k(y)\| > r$, or equivalently,

$$\|f_k(x) - f_k(y)\| \leq r \text{ implies } \varrho(x, y) \leq 2r.$$

Thus, intuitively, the smaller is r , the “intermediate function” f_k is “more injective”. According to step 5, ε' can be regarded as an estimation of “non-injectivity” of f_k and the algorithm is keep going whenever the value of ε' exceeds ε (the tolerable amount of inaccuracy given in the input). At the end we have

$$(*) \text{ if } \|f(x) - f(y)\| \leq \varepsilon/2 \text{ then } \|x - y\| \leq \varepsilon.$$

Further, by Lemma 1 of [9], for all k and $x, y \in \mathbb{R}^n$ we have

$$\|f_k(x) - f_k(y)\| \leq \sqrt{k} \cdot \|x - y\|,$$

particularly,

$$(**) \text{ if } \|f_k(x) - f_k(y)\| > \sqrt{k}\varepsilon \text{ then } \|x - y\| > \varepsilon.$$

So $(*)$ and $(**)$ can be summarized in the following three cases: let $x, y \in \mathbb{R}^n$ be arbitrary and let $d := \|f(x) - f(y)\|$.

- if $d \leq \varepsilon/2$ then $\|x - y\| \leq \varepsilon$;
 - if $d > \sqrt{k}\varepsilon$ then $\|x - y\| > \varepsilon$;
 - if $\varepsilon/2 < d \leq \sqrt{k}\varepsilon$ then we have to compute $\|x - y\|$ in order to determine whether $\|x - y\| \leq \varepsilon$.
- This is what we are doing in the Searching part.

As A is a finite set, the Initializing part always terminates. In fact, because of Step 2, it terminates after at most n many iterations of steps 3-6. If $k = n$ after Initializing, then this method is unable to reduce the dimension.

It is straightforward to see, that the number of steps of the Initializing part is proportional with $|A|^2$ in the worst case. The precise number of required steps strongly depends on ε and the structure of A , hence, at that level of generality we cannot improve the estimation for the time complexity of the Initializing part. However, we conjecture, that in particular situations, by a careful choice of ε , the number of required steps may be kept in a reasonably small level. We are planning to implement and test our algorithm on real life databases (such investigations are in progress at the moment).

Remarks on the Searching part. It may happen, that there are several $\langle u, v_0 \rangle, \dots, \langle u, v_m \rangle \in B$ for which $\|f(x) - u\| \leq \frac{\varepsilon}{2}$ (where x is the input).

According to the choice of k in the Initializing part, we have $\varrho(v_i, v_j) \leq 2r \leq \varepsilon$ for all $i, j < m$, where $r = \max_{x \in A} \varrho(\{a_j : j \leq k\}, x)$.

The number of steps in the Searching part is proportional with $|A|$, but when we compute $\|f(x) - u\|$, we are using the distance function of the k -dimensional Euclidean space. As k may be substantially smaller than n , this method may be more efficient than checking the elements of A step-by-step with the distance function of the n -dimensional Euclidean space. Further, because the Searching part is essentially a minimum-searching problem, it seems possible to accelerate Step 1 further by applying well known methods of algorithm theory or operation research.

III. CONCLUSION

We proposed a similarity detecting algorithm. The similarity between objects had been described by a metric ϱ . Our method is based on two parts: the Initializing and the Searching parts. In the special case, when the metric ϱ is the usual Euclidean distance of a large dimensional Euclidean space \mathbb{R}^n , our method can be considered as a way of reducing dimension. In this particular case we analyzed the time complexity of the Initializing and the Searching parts.

In the future we are planning to implement our algorithm and test it in real life databases.

ACKNOWLEDGMENT

This work has been supported by Hungarian National Foundation for Scientific Research grant K129211.

REFERENCES

- [1] Engelking, R., *General Topology*, Heldermann Verlag, Berlin, (1989).
- [2] Faloutsos, C. and Lin, K., *FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, ACM., Vol. 24, No. 2, 163–174, (1995).
- [3] R. J. FOWLER, M. S. PATERSON AND S. L. TANIMOTO, *Optimal packing and covering in the plane are NP-complete*, Information Processing Letters, Vol. 12, No. 3, 133-137, (1981).
- [4] T. F. GONZALEZ, *Clustering to minimize the maximum intercluster distance*, Theoretical Computer Science, 38: 293-306, (1985).
- [5] S. HAR-PELED, *Geometric Approximation Algorithms*, American Mathematical Society, Boston, US, (2011).
- [6] Hjalton, G.R. and Samet, H., *Contractive embedding methods for similarity searching in metric spaces*, Technical report, Computer Science Department, Center for Automation Research, Institute for Advanced Computer Studies, University of Maryland, (2000).
- [7] V. Marianov and H. A. Eiselt editors, *Foundation of Location Analysis*, Springer Verlag, US, (2011).
- [8] J. R. Munkres, *Topology*, Prentice Hall, US, (2000).
- [9] G. SÁGI, *Almost injective Mappings of Totally Bounded Metric Spaces into Finite Dimensional Euclidean Spaces*, Advances in Pure Mathematics, 9. pp. 555-566 (2019).
- [10] G. Sági , and D. Nyiri, *On embeddings of finite metric spaces*, in: the Proceedings of the 13th International Scientific Conference on Informatics (editors: V. Novitzka, S. Korečko and A. Szakál), IEEE, (2015).
- [11] J. Satyabrata and P. Supantha, *Covering and packing of rectilinear subdivision*, WALCOM: algorithms and computation, 381Ú393, Lecture Notes in Comput. Sci., 11355, Springer, Cham, (2019).
- [12] R. Zahed and T. M. Chan, *A clustering-based approach to kinetic closest pair*, Algorithmica Vol. 80, No. pp. 10, 2742Ú2756, (2018).

A Deep Meta-learning Framework for Heart Disease Prediction

Iman Salem

Arab Academy for Science, Technology and
Maritime Transport (AASTMT),
Alexandria, Egypt
imanmahdi89@gmail.com

Radwa Fathalla

Arab Academy for Science, Technology and
Maritime Transport (AASTMT),
Alexandria, Egypt
radwa_fathalla@aast.edu

Mohamed Kholeif

Arab Academy for Science, Technology and
Maritime Transport (AASTMT),
Alexandria, Egypt
Kholeif@gmail.com

Abstract—Heart disease is a leading cause of death. This is proved by the high mortality rates annually published. The factors and symptoms of this disease which are significantly common among other diseases, grabbed the attention of the scientific community to integrate advanced tools and computing systems such as Machine Learning (ML), and Artificial Neural Networks (ANN) to early diagnose and predict this disease. Among recent research work, Restricted Boltzmann Machine (RBM) formed a viable model to solve similar problems in the medical domain. Therefore, we introduce this paper to present a heart disease classification system based on RBM. In this paper, we relied on a standard dataset, namely the Cleveland dataset to conduct our experimental work. In addition, we developed a meta-learning framework to train stacked RBM classifiers, where we use the same type of classifier with different levels of abstractions of the data to get several classifications. The experimental results show that our proposed model achieves an accuracy that is on par with the state-of-the-art.

Keywords— Heart disease, Restricted Boltzmann Machines, Meta-Learning, Multilayer Perceptron.

I. INTRODUCTION

Heart disease is a form of cardiovascular disease that directly affects millions of citizens' lives around the world. The survey analysis of the World Health Organization (WHO) shows about 17 million death cases are due to heart disease [1]. In addition, according to the Institute for Health Metrics and Evaluation (IHME), during the period from 1990 to 2013, the death rate has increased by 41% due to this disease. Further, in 2014, this rate has shown an alarming of 9.83% further increase [2]. This increased prevalence and mortality have necessitated intervention in the form of developing smart classification tools. These tools can aid in differential diagnosis carried out by the medical practitioners.

The most common predisposing factors that contribute to the heart disease may include inheritance problems, cholesterol, high blood pressure, poor diet, high calcium rate, and narrowed blood vessels. In addition, the symptoms of heart disease may include discomfort, pressure, heaviness, or pain in the chest, arm, or below the breastbone. These factors

and symptoms significantly overlap with many other diseases, which make heart disease diagnosis a challenging endeavor for any physician.

These challenges were the motivation for the scientific community to exploit modern tools and computing systems such as Machine Learning (ML), Data Mining (DM), Artificial Intelligent (AI), to early diagnose and predict heart disease [3, 4, 5]. The main goals for this exploitation was to process complex data inputs from many sources, analyze these data, then recognize certain patterns to easily and early define the disease.

In addition, in the last few years, Restricted Boltzmann Machine (RBM) [6], a generative stochastic Artificial Neural Network (ANN) was introduced as a viable model to solve a range of problems in various domains, such as dimensionality reduction [7], image classification [8], recommender systems [9], medical services [10], and in the context of clustering [11]. RBM was introduced to define the probability distribution over primarily binary-valued patterns [12]. Recently, RBM played a major role in Deep Learning frameworks. This model was successfully implemented to solve various sophisticated problems especially in the medical domain [13, 14].

Further, RBMs were found to be specifically useful as the dataset contain some missing values for some attributes such as age, smoking rate, diuretic, among others. RBMs were proven to effectively handle missing data in the research literature up to 40% of record data missing [10, 15, 16]. In our work, RBMs managed to handle this situation to avoid much false scenarios and contribute in recognizing true heart disease cases.

Traditionally, machine learning techniques relying on classical metadata learning algorithms (learning-from-learning approaches) utilize different types of classifiers to yield various decisions. Then, the outcomes are combined using a merging rule, usually a majority vote. For example, Louisa et al. [17] demonstrated that combining the decisions of several classifiers, can lead to better recognition result. Such combination can be implemented using a variety of

strategies, among which majority vote is by far the simplest, and yet it has been found to be just as effective as more complicated schemes in improving their recognition results. In addition, Peng et al. [18] discussed the accuracy of classifier combining based on majority voting. They formulated the accuracy with only two parameters; the average accuracy of individual Classifiers and the efficiency of the classifiers combined as Lapsed Accuracy (LA).

However, in this paper, we introduce the use of same classifier, the RBM fed with different levels of abstractions to perform this task and gain more accurate and efficient learning. Our recommended merging rule utilizes a Multi-Layer Perceptron (MLP) machine.

We introduce this paper to contribute to the diagnosis of heart diseases based on RBM. In this paper, we relied on the Cleveland benchmark dataset [19] to conduct our simulation work and demonstrate the effectiveness of the proposed solution.

The rest of this paper is organized as follows; we briefly introduce other related work in section II. Section III presents the proposed classification system based on RBM. Section IV shows our experimental work and results. In addition, this section provides a comparison between our results and other research works. Finally, Section V concludes our research study and presents the suggestions for future work.

II. RELATED WORK

This section briefly presents the most related research that addressed the use of RBM solutions in the medical field. In addition, it includes other machine learning solutions for the diagnosis of heart diseases and using same Cleveland dataset.

RBM deployment. Sun et al. [20] proposed to use the Deep Restricted Boltzmann Machine (DRBM) to replace the traditional classifiers used in the area of radiomics. DRBM is a stack of multiple RBMs. It is an undirected network of interconnected hidden layers with the capability to learn joint probabilities over these connections. The evaluation results were presented in terms of Receiver Operating Characteristic (ROC) [21] and Area Under Curve (AUC). Their experimental results show that the AUC of DRBM was increased by about 19% compared to traditional logistic regression and support vector machine methods, when tested on 7 datasets of CT image scans.

Tulder et al. [15] introduced a convolutional classification RBM, which combines a generative and a discriminative learning objective. The convolutional RBM uses the convolutional weight-sharing pattern from convolutional networks to learn small filters that are applied to every position in large HRCT image series with three-dimensional annotated regions of pathological lung tissues.

This weight sharing makes learning more efficient and allows the RBM to model small features that occur in multiple areas of an image, which is useful for describing textures. Tulder presented experiments with feature learning for lung texture classification and airway detection in CT images. His experimental results showed an increase in the lung tissue classification accuracy by 1 to 8 percentage points.

Kim et al. [10] proposed a concept of emergency detection algorithm for healthcare robot which adopts discriminative RBM for anomaly detection in audio and video recordings collected for at-home patients. He adopted anomaly detection rather than simple emergency case classification as it is hard to collect real emergency data to train an effective classifier. However, they only provided the novel idea for emergency detection without conducting any experimental work.

Heather et al. [16] introduced classification of respiratory disturbances in Rett Syndrome (RTT) [22] patients using RBM. Heather utilized temporal, flow, and autocorrelation features taken from the respiratory inductance plethysmography chest signal, which are digital waveforms for the movement of the chest and abdominal wall. He tested the performance of six classifiers including; Support Vector Machine (SVM-R) [23], RBM, a Levenberg-Marquardt NN Classifier (LMNC) [24] and a Back-Propagation NN Classifier (BPNC) [25]. To evaluate the performances of these classifiers, they propose two different modalities: (1) a Leave-One-Subject-Out (LOSO) [26] for evaluating the variability inter-subject and (2) a Leave-One-Trial-Out (LOTO) [27] per subject to evaluate how independent each classifier is in terms of each individual respiratory event. Their experimental work showed that, in case of the LOSO modality, RBM and Decision-Fusion classifiers maintain F1-scores above 88%. This means semi-supervised classifiers can robustly classify respiratory events when the feature set includes a combination of such features. In addition, RBMs outperformed all classifiers in both modalities.

Heart disease prediction. On another front, several computing solutions have been introduced for the heart disease investigation problem.

Mehmood et al. [2] introduced the prediction of heart disease using ANN. They relied on the Principle Component Analysis (PCA) [28] for the dimensionality reduction process. In addition, they explore a number of ANN techniques to compare the results before and after the PCA process. Their experimental work demonstrated that, the accuracy of the prediction has risen from 94.7% to 97.7% with the inclusion of PCA.

Tomov et al. [29] investigated the potential of Deep Neural Networks (DNNs) for detecting heart disease based on routine clinical data. DNN consists of more than the traditional two layers. It uses sophisticated mathematical

modeling to process data in complex ways. The novel work led to the construction of a new five-layered architecture for DNN. These layers were named as the Heart Evaluation for Algorithmic Risk-reduction and Optimization Five (HEARO-5). In addition, Tomov et al. relied on k-fold cross-validation as well as Matthews Correlation Coefficient (MCC) [30] to measure the quality of the classifications. Further, her experimental work showed that the HEARO-5 architecture, yielded about 99% of detecting accuracy and 0.98 as the MCC figure, which significantly outperforms other ANN solutions.

Jabbar et al. [5] proposed a classification and a feature subset selection method of heart disease using ANN as a preprocessing step to reduce dimensionality and remove irrelevant data. This research work was introduced to determine the attributes which contribute more towards the predication of heart disease which indirectly reduces number of diagnosis tests that are needed to be taken by a patient. Their experimental results showed that the prediction accuracy is much improved over traditional classification settings.

Amma et al. [31] presented a system for heart disease diagnosis by combining ANN and a genetic algorithm. In this research work, preprocessing the dataset consisted of filling out missing values and normalizing the data using Min-Max normalization. Their repository of patients relied on 12 attributes of nominal and numeric types. Additionally, the weights of the neural network were determined using the genetic algorithm and the accuracy obtained from these study was 94.17%.

Research work reported in [2, 29, 31] used the Cleveland dataset [19] to conduct experiments as our research. It consisted of categorical, integer, and real-valued data.

It is important to note that, various computing system techniques were employed for different heart disease related applications. However, we clearly noted the lack of using the RBM for the diagnosing and classification of heart diseases. Therefore, we introduce this paper to contribute in exploiting RBM for the diagnosing of this disease.

III. THE PROPOSED RBM-BASED CLASSIFICATION SYSTEM

The proposed model is a stacked RBM model. It leverages the power of RBM to learn the associations between the record classes and the features of the patients, and to solve the problem of missing data imputation. In addition, a meta-learning decision based on a Multilayer Perceptron (MLP) model fed with a sufficient number of a combined RBM classifications is adopted to gain more accurate and efficient learning.

A. Restricted Boltzmann Machine (RBM)

Restricted Boltzmann machines (RBM) is an energy-based undirected model that uses a layer of hidden variables to model a distribution over visible variables [32]. There are two learning models in which RBMs are exploited; the Generative model and the Discriminative model.

The Generative models learn to represent the Probability Density Functions (PDF) using a set of data known as the training data. This model learns to represent the entire set of training data which for the most part include some unlabeled data [33]. It has been mainly used as a generative model of many different types of datasets containing labeled or unlabeled data [34] and is mainly used for reconstruction of the data. Whereas, Discriminative models learn to classify training data by calculating the posteriors, where the true classes are pre-known and supplied in the ground truth data as input to the training [35].

RBM is a Markov Random Field (MRF) [36] associated with an undirected bipartite graph $G = (V, H)$; where “v” are the visible units as $v \in \{0, 1\}^v$ and “h” are the hidden units as $h \in \{0, 1\}^h$. The special bipartite structure separating these two kinds of units into layers allows each unit to be only dependent on the other layer’s units. Fig. 1 illustrates RBM model with two bipartite layers consisting of “x” and “y” fixed number of units per layer.

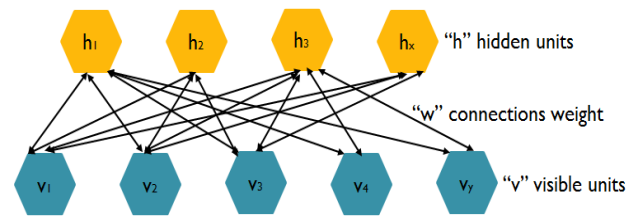


Fig 1. Restricted Boltzmann Machine (RBM)

Therefore, joint configuration $E(v, h)$ of the visible and hidden units has an energy that can be calculated as follows;

$$E(v, h) = -\sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i, j} v_i h_j w_{i, j} \quad (1)$$

Where v_i , h_j are the binary states of visible unit i and hidden unit j , a_i , b_j are their biases and w_{ij} is the weight of the connections between them.

This network assigns a probability to every possible pair of a visible and a hidden vector using the following energy function:

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (2)$$

Where the Z is the “partition function” that is given by summing over all possible pairs of visible and hidden vectors:

$$Z = -\sum_{v,h} e^{-E(v,h)} \quad (3)$$

The conditional probabilities are given by;

$$P(h|v) = \prod_{j=1}^y P(h_j = 1|v) \quad (4)$$

and

$$P(v|h) = \prod_{j=1}^x P(v_j = 1|h) \quad (5)$$

In addition, the probability of a single unit being active can be defined as;

$$P(h_j = 1|v) = \sigma(\sum_{i=1}^y v_i w_{ij} + c_i) \quad (6)$$

and

$$P(v_j = 1|h) = \sigma(\sum_{i=1}^x h_i w_{ij} + c_j) \quad (7)$$

Where c_i, c_j are the bias values for the visible and hidden nodes respectively and $\sigma(x) = \frac{1}{1+e^{-x}}$

Furthermore, the probability that the network assigns to a visible vector, v , is given by summing over all possible hidden vectors as follows:

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \quad (8)$$

This leads to a very simple learning rule to perform stochastic steepest ascent in the log probability of the training data:

$$\Delta w_{i,j} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \quad (9)$$

Where ϵ is a learning rate

Approximations to the energy function led to a fast learning procedure proposed by Hinton in 2002 [37], namely Contrastive Divergence (CD). This starts by setting the states of the visible units to a training vector. In this case, the binary states of the hidden units are all computed in parallel using equation 6. Once binary states have been chosen for the hidden units, a “reconstruction” is produced by setting each v_i to 1 with a probability given by equation 7.

Therefore, the change in a weight can be approximately calculated as follows;

$$\Delta w_{i,j} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \quad (10)$$

B. Stacked RBMs in a Meta-Learning Framework

In this paper, we propose a classification RBM, which combines both the generative and the discriminative learning objectives. The proposed model consists of five-layered stacked architecture of standalone RBMs.

Traditionally, in the classification applications RBM layers are used in a pre-training setting [32]. Contrastive

divergence (CD) [37] is used to jointly learn latent representations of the input features on the visible nodes. Afterwards, the layers are unfolded and complemented with an output layer of the data classes. Then, the architecture is further trained using Multi-Layer Perceptron Back-Propagation. The whole framework is referred to as discriminative use of RBMs. However, in our work, we use each single-layered standalone RBM to jointly learn the probability distribution of the input features and the ground truth classifications in a purely generative setting and using only CD.

Initially, to learn the model parameters, we supply the generative RBM visible nodes with the dataset training records. Despite the fact that, the standard RBM is an unsupervised model, the proposed RBM extends the standard RBM by adding a set of label nodes to the visible layer as shown in figure 2. This allows the RBM to learn the joint probability of the input, the hidden representation, and the label. Further, the supervised learning procedure periodically extracts some features to generate the latent representation on the hidden nodes of each RBM layer. The representation is further augmented with the ground truth labels and fed into the next RBM layer (in an upward direction) as input by clamping the resulting vector to its visible nodes.

The proposed model comes up with an appropriate strategy to handle missing data during both the training and testing phases. We managed to impute a “zero” value in place of all missing attributes and in place of the labels in the testing. This process is quite accurate compared to other NN data imputation techniques [38, 39, 40]. This quality of the RBMs, the ability to handle missing values, has been verified in multiple research [41]. Several authors proved that RBMs can handle up to 40% of the count of inputs missing without affecting the performance. It is an important quality as it alleviates the need to interpolate/extrapolate values to substitute for the missing ones. In some research, the problem even goes as far as completely eliminating the records or attributes with missing values.

In addition, the proposed model maintains the use of the same RBM classifier at different level of abstraction to perform a sort of meta-learning solution. The latent representations formed at the hidden nodes of the layered RBMs transforms the raw data into other feature spaces. The learned parameters, weights and biases, constitute the kernel functions that do the transformation. Thus, the meta-learning is achieved by examining the classifications of the same type of classifier, namely RBM, on the data in various feature spaces rather than examining the outcomes of different classifiers operating on the raw data.

During the testing phase, the model is applied to previously unseen records (the testing data) where the labels in the ground-truth are omitted at the input. The process of creating sequential sets of activations by grouping features/labels forms the basis of a node hierarchy, by which

the proposed model tests a complex and abstract representations of data. Therefore, the proposed network is handled in a greedy, layer-by-layer manner, where the bottom layer is tested alone as an RBM, and then the values at its hidden nodes are clamped to the visible nodes of the next layer, and so on. In addition, each layer makes a vote for each unlabeled test instance by reconstructing the data on the visible nodes in place of the omitted labels. The levels of activations on these reconstructed nodes indicate the probabilities of the different labels. Thus, the label with the maximum probability is chosen at each layer. And the final output prediction is the one that receives the maximum voting count from various layers.

In another mode of determining final classification, we relied on the Multilayer Perceptron (MLP) technique to feed-forward the proposed model and generate a set of five outputs from a set of inputs gathered from the RBM layers. We trained the MLP with the labels, resulting at the end of the training as reconstructions on the visible nodes of 5 layered RBMS, thus comprising a set of 25 probability values, against the groundtruth labels on the output layer of the MLP. In this way the MLP was trained to yield the final classification using the probabilities resulting from the RBM reconstructions. Further, during the testing phase, the reconstructions of the unseen data are once again aggregated and fed into the MLP to achieve the final prediction.

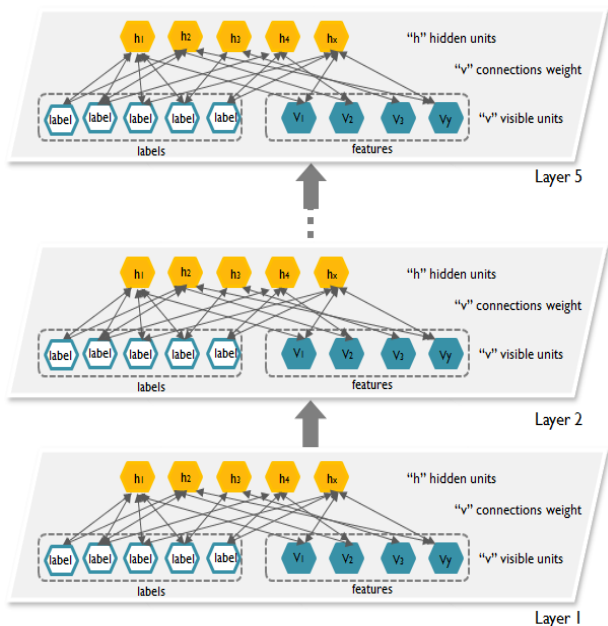


Fig 2. The Proposed Stacked RBM Model

IV. EXPERIMENTAL WORK

A. Data Preprocessing and Experimental Setup

The adopted Cleveland benchmark dataset include about 303 instances with 76 attributes. In this paper, we managed

to use about 93% of these instances to conduct our experimental studies.

In addition, most of the aforementioned research works [2, 29, 31] relied on 14 attributes only to conduct their experimental studies, again, due to the existence of missing values and irrelevance of some attributes (such as name, social security number, patient identification number, etc.). On the contrary, in this paper, due to the high tolerance of our RBMs to missing data, we have discarded the irrelevant attributes or the ones that suffer from very high degree of sparsity in a way that may degrade system performance. Therefore, we managed to use a whole of 39 attributes to demonstrate the effectiveness of the proposed model.

In addition, the main attribute is the "goal" field which refers to the presence of heart disease in the patient. This field is ranged between five integer values starting from "0" to "4". Each of these indices refers to the existence and severity of the disease. The dataset is divided as follows: 157 of absent disease, 50 low severity, 31 medium severity, 32 high severity, and 12 serious cases.

These 39 attributes include two types of values; first the Categorical values (such as male/female), and the second is the numeric continuous values. For the former, the categories are encoded as integer values. As for the later, we rely on the "Equal-Width" binning processing technique to map these values to integers representing the bin number. In either cases, these integer values are encoded in a one-hot vector allowing the values to be in the range of 0s and 1s so that it can be clamped to the binary visible nodes of the RBM. Same handling goes for the ground truth labels, dealing with them as categorical data. Thus, converting each label to a 5-elements one-hot vector, where 1 is placed on the node corresponding to the label number and 0s otherwise.

Furthermore, RBM is a stochastic ANN. It uses randomness while being fit on any dataset, such as random initial weights and random shuffling of data. Therefore, to discover how many visible and hidden nodes are required in the RBMs, we ran many pilot tests during the systematic experimentation phase. The results of these experiments showed that the optimal number for both visible and hidden nodes is 217. This value guarantees the best error reduction of the prediction process. Equality of nodes count between visibles and hiddens is supported in related papers in the work of Hinton [37].

We utilized the 5-fold cross-validation technique [42] to evaluate the proposed model by partitioning the selected instances into a training set to train the model, and a test set to evaluate it. In our 5-fold cross-validation, the selected instances is randomly partitioned into 5 equal size subsets of samples. Of the five subsets, a single sub group is retained as the validation data for testing the model, and the remaining 4 subsets are used as training data. The 5-fold cross-validation process is then repeated 5 times, with each

of the 5 subsamples used exactly once as the validation data. The advantage of this technique is that all observations are used for both training and validation, and each observation is used for validation exactly once.

The outcomes of the stacked independent RBMs were combined in the Meta Learning framework. One approach was taking the majority vote of the 5 layers. In case of a tie, we choose the class with the higher average score.

Another approach was topping the architecture of stacked RBMs with a MLP. After the RBMs have been trained, we ran them on the same seen data but this time while suppressing the true classes. The reconstructed probabilities are aggregated of the 5 layers of RBMs and fed into a MLP. The MLP with 1 hidden layer is trained for 1000 iterations, where the labels are the 5 classes in the ground truth. At test time, the MLP operates on the probabilities reconstructed for the unseen data. The outcome on the output layer of the MLP is considered the final decision.

B. Results and discussion

As mentioned earlier, the Cleveland Heart Disease Dataset is used for training and testing our proposed solution based on 5-fold cross-validation.

In addition, table I demonstrates the proposed model accuracy results using Cleveland dataset. We relied on the accuracy measure to demonstrate the effectiveness of our proposed model. This accuracy is calculated as the average number of the true positive records (the correctly instances predicted by the model when compared to the ground truth data) over the 5-folds normalized by the number of instances in the testing fold.

TABLE I
The Proposed Model Accuracy Results

	Our proposed model Results		
	V ₁	V ₂	V ₃
Accuracy	94.83%	92.24%	96.55%

In this table, we include the following 3 variants of our model:

V₁: A single layered RBM.

V₂: A meta-learning decision based on the majority vote of 5 stacked RBMs.

V₃: A meta-learning decision based on a MLP model fed with the output probabilities of 5 stacked RBMs.

It is clear that our best performance was obtained using the MLP meta-learner. This is due to the fact that the MLP is able to deduce more complex intrinsic relations among the suggested class probabilities resulting from the RBMs, than a simple majority vote.

The standard deviation for the accuracy of V₃ model was found to be ± 2.13 across the 5 folds. This, if compared with others, indicates the stability of the performance of our model. But unfortunately, only mean accuracy is reported in peer research. Mehmood [2], Tomov [29], and Amma [31] reached an accuracy of 97.9%, 99%, and 94.17% respectively using same dataset. They relied on traditional machine learning techniques that required extensive data preprocessing steps that reduced the set of utilized attributes to 14 due to the inability of such techniques to deal with missing data. In addition, traditional techniques do not support fully-automated processing and require the manual setting of some parameters. In all our model variants, RBM is a prior dependent technique, in the sense that it reconstructs a lower probability to the less frequently encountered classes. Thus, the misclassifications were mostly seen in the “serious cases” group due to its under representation in the dataset. RBM layers were incrementally added to the stack and with more layers piled, the performance of the majority voting mode deteriorated as can be detected by V₁ versus V₂ accuracies in table I. However, the results in the MLP mode has been enhanced with the addition. This indicates it is the pattern of classifications that is crucial to decision making rather than the individual values.

A problem seen in similar settings as ours with small-sized datasets is overfitting, in which accuracy on seen data increases and on unseen data decreases. We mainly attribute our model competitive performance, despite the size limitation, to the binning concept used in converting the data values into one-hot-vector. It prevents the model from memorizing the data as exact values, instead it learns the data in the form of ranges. This greatly helped in boosting the generalization ability of the model. Luckily, the nature of health investigations lends itself to such mechanism because they are examined by medical personnel by their ranges not exact values. In a sense, binning is equivalent to adding noise to data records during training to deviate it from exact values, a well-known technique to combat overfitting [43]. Other factors that reduced the impact of overfitting include the following. The 5-fold cross-validation allowed the hold out part used for evaluating the model to be relatively small, sparing most of the available data instances for training, a good handling for small datasets. And, at the same, taking the average over several rounds of training allowed it to be a realistic statistic of performance. In addition, in our pilot experiments, we worked on optimizing some hyperparameters of the model. We settled on utilizing a weight regularizer by adding a L1-penalty to bias coefficients towards small values. Large weight values is a property associated with a model that overfits on training data. Another parameter which affects the convergence of the model to optimal/suboptimal minima is the batch size, by directly affecting the number of updates per epoch. The trade off is between using a small batch, introducing noise to the model but better exploring the solution space versus using a large batch which is a better representative of the dataset

leading to earlier convergence but possibly to a local minimum in the solution space of a lower grade. We chose 28 instances based on trial and error. We exploited the early stopping tactic to avoid over-training. We would like to point out that the relatively early stopping at 50 epochs was facilitated by the placement of labels on the visible nodes. This led to the concurrent learning of features and labels. This is in contrast to the common arrangement that performs the learning in 2 phases [32], pretraining using CD followed by fine-tuning by backpropagation, which all-in-all requires more training epochs. Finally, it is important to note that overfitting is exaggerated by model complexity and stacking layers does not add to the complexity of the model in the same sense as other deep learning approaches such as Convolution Neural Networks (CNN), because each layer is trained separately. So, there is no coupling across layers with respect to parameters update. In other words, we can not add up the number of free parameters of all RBMs to express complexity, as each layer is a standalone model contributing equally with others to the final decision.

V. CONCLUSION

The tremendous increase in death rate due to cardiovascular and heart diseases motivated the scientific communities to exploit modern computing techniques for early detection of the disease. Building a meta-learning model based on stacked RBMs leads to exploring complex interactions between data attributes in various feature spaces. This led to more robust decision making. Therefore, in this paper, we present an architecture composed of 5 stacked RBMs used to learn a joint probability distribution of heart disease markers and labels referring to disease severity. A MLP was used to leverage the accuracy of the final classification. It was better able to study the patterns of label probabilities obtained from the reconstructions of the RBMs, rather than the simple majority vote approach. We demonstrate the efficacy of our proposed model on the Cleveland benchmark dataset based on 5-fold cross-validation experiments. The results prove the competitive performance. The main merit of the work, it is field independent. The technique is generic enough to be applied in other diagnostic settings and datasets. Future work could consider incorporating Magnetic Resonance Imaging (MRI) data to build a more comprehensive diagnosis system by performing advanced image classification tasks.

REFERENCES

- [1] Organization, Geneva, Switzerland: World Health Organization, 2014.
- [2] S. Mehmood, M. Usama, and A. G. Khan, "Prediction of Heart Disease using Artificial Neural Network," *VFAST Trans. Softw. Eng.*, no. October, pp. 102–112, 2018.
- [3] T. Marikani and K. Shyamala, "Prediction of Heart Disease using Supervised Learning Algorithms," *Int. J. Comput. Appl.*, vol. 165, no. 5, pp. 41–44, 2017.
- [4] S. Babu, E. M. Vivek, K. P. Famina, K. Fida, P. Aswathi, M. Shanid, and M. Hena, "Heart disease diagnosis using data mining technique," *Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017*, vol. 2017-January, pp. 750–753, 2017.
- [5] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, "Classification of Heart Disease using Artificial Neural Network Classification of Heart Disease using Artificial," *Glob. J. Comput. Sci. Technol. Neural Artif. Intell.*, vol. 13, no. 3, pp. 5–14, 2013.
- [6] G. Montúfar, "Restricted boltzmann machines: Introduction and review," *Springer Proc. Math. Stat.*, vol. 252, no. October, pp. 75–115, 2018.
- [7] G. Hinton, and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, 504, 2006.
- [8] K. Sohn, D. Yon Jung, H. Lee, and A. O. Hero, "Efficient Learning of Sparse, Distributed, Convolutional Feature Representations for Object Recognition," *IEEE International Conference on Computer Vision*, 2643–2650, 2011.
- [9] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering", *the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.
- [10] H. G. Kim, S. H. Han, and H. J. Choi, "Discriminative restricted Boltzmann machine for emergency detection on healthcare robot," *2017 IEEE Int. Conf. Big Data Smart Comput. BigComp 2017*, pp. 407–409, 2017.
- [11] N. A. Huhnstock, A. Karlsson, M. Riveiro, and H. J. Steinhauer, "On the behavior of the infinite restricted boltzmann machine for clustering," *ACM Symposium on Applied Computing*, pp. 461–470, 2018.
- [12] T. Osogami, "Boltzmann machines and energy-based models," *IJCAI-17 tutorial on energy-based machine learning*, pp. 1–36, 2017.
- [13] L. Ning, R. Pittman, and X. Shen, "POSTER: Cutting the Fat: Speeding Up RBM for Fast Deep Learning Through Generalized Redundancy Elimination," *Parallel Archit. Compil. Tech. - Conf. Proceedings, PACT*, vol. 2017-September, pp. 154–155, 2017.
- [14] C. L. P. Chen, C. Y. Zhang, L. Chen, and M. Gan, "Fuzzy Restricted Boltzmann Machine for the Enhancement of Deep Learning," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2163–2173, 2015.
- [15] G. Van Tulder and M. De Bruijne, "Combining Generative and Discriminative Representation Learning for Lung CT Analysis With Convolutional Restricted Boltzmann Machines," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1262–1272, 2016.
- [16] H. M. O'Leary, J. M. Mayor, C. S. Poon, W. E. Kaufmann, and M. Sahin, "Classification of respiratory disturbances in Rett Syndrome patients using Restricted Boltzmann Machine," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 442–445, 2017.
- [17] L. Lam and C. Y. Suen, "Application of Majority Voting to Pattern Recognition: An Analysis of its Behavior and Performance," *IEEE Trans. on systems, man, and cybernetics—part a: systems and humans*, vol. 27, no. 5, 1997.
- [18] P. Hong, L. Chengde, L. Linkai, and Z. Qifeng, "Accuracy of classifier combining based on majority voting," *2007 IEEE Int. Conf. Control Autom. ICCA*, vol. 00, pp. 2654–2658, 2008.
- [19] Available [Online]: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>
- [20] C. Sun et al., "Non-Small Cell Lung Cancer Staging Using Deep Restricted Boltzmann Machine," *Int. Conf. Signal Process. Proceedings, ICSP*, vol. 2018-August, pp. 1175–1178, 2019.
- [21] J. Hernández-Orallo, "ROC curves for regression," *ELSEVIER Pattern Recognit.*, vol. 46, no. 12, pp. 3395–3411, 2013.
- [22] S. M. Kyle, N. Vashi, and M. J. Justice, "Rett syndrome: a neurological disorder with metabolic components.," *Open Biol.*, vol. 8, no. 2, 2018.
- [23] A. Skomorokhov and A. Nakhbov, "Support vector machines in A+," *ACM SIGAPL APL Quote Quad*, vol. 34, no. 4, pp. 8–17, 2007.
- [24] M. Lourakis, "A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar," *Tech. Report-Institute Comput. Sci.*, vol. 3, no. January 2005, p. 2, 2005.
- [25] M. Buscema, "Back propagation neural networks," *Subst. Use Misuse*, vol. 33, no. 2, pp. 233–270, 1998.
- [26] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, Vol. 4, 2010.

- [27] Montúfar, Guido. "Restricted Boltzmann Machines: Introduction and Review." *Information Geometry and its Applications IV*. Springer, Cham, 2016.
- [28] K. L. Sainani, "Introduction to principal components and factor analysis," *PM R*, vol. 6, no. 3, pp. 275–8, 2014.
- [29] N.-S. Tomov and S. Tomov, "On Deep Neural Networks for Detecting Heart Disease," *Journal of Artificial Intelligence in Medicine*, 2018.
- [30] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS One*, vol. 12, no. 6, pp. 1–17, 2017.
- [31] N. G. B. Amma, "Cardiovascular disease prediction system using genetic algorithm and neural network," *2012 Int. Conf. Comput. Commun. Appl. ICCCA 2012*, 2012.
- [32] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [33] S. Nie and Z. Wang, "A Generative Restricted Boltzmann Machine Based Method for High-Dimensional Motion Data Modeling," *ELSEVIER Computer Vision and Image Understanding*, Volume 136, Pages 14-22, 2015.
- [34] Hinton G. E., Osindero S., and The. Y. W., "A fast learning algorithm for deep belief networks", *Neural Computation*, 18(7):, 1527-1554, 2006.
- [35] H. Larochelle, "Classification using Discriminative Restricted Boltzmann Machines," *the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.
- [36] A. Fischer , "Training Restricted Boltzmann Machines," *Springer, KI - Künstliche Intelligenz: Vol. 29, No. 4*. pp. 441-444, 2015.
- [37] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [38] G. Chhabra, V. Vashisht, and J. Ranjan, "A classifier ensemble machine learning approach to improve efficiency for missing value imputation," *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 23–27, 2019.
- [39] Si, "Intelligent imputation technique for missing values," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 2441–2445, 2016.
- [40] Si, "Proper imputation techniques for missing values in data sets," *Proc. 2016 Int. Conf. Data Sci. Eng. ICDSE 2016*, 2017.
- [41] Y. Sakai and K. Yamanishi, "Data Fusion Using Restricted Boltzmann Machines," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2015-January, no. January, pp. 953–958, 2015.
- [42] T. Gunasegaran and Y. N. Cheah, "Evolutionary cross validation," *ICIT 2017 - 8th Int. Conf. Inf. Technol. Proc.*, pp. 89–95, 2017.
- [43] You, Zhonghui & Ye, Jinmian & Li, Kunming & Wang, Ping. "Adversarial Noise Layer: Regularize Neural Network By Adding Noise," *Proc. International Conference on Image Processing, ICIP 2018*, 2018.

epiBAT: Multi-objective bat algorithm for detection of epistatic interactions *

Jozef Sitarčík

*Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
xsitarcik@stuba.sk*

Mária Lucká

*Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
maria.lucka@stuba.sk*

Abstract—Detection of epistatic interactions associated with diseases can improve prevention and diagnosis of those diseases. Epistatic interactions are nonlinear interaction effects of single nucleotide polymorphisms (SNPs), which are substitution mutations occurring at some specific position in the genome. Detecting associations between them is very computationally expensive, as more complex diseases can be associated only with epistatic interactions of two and more SNPs, thus making a very large quantity of possible SNP combinations needed to test. To cope with such high computational complexity, current methods are also based on bio-inspired algorithms. In this paper we propose epiBAT, a new algorithm based on bat algorithm with multiple objectives and tabu search. We apply our algorithm on different testing data sets and compare it with other existing methods. The experiments have shown that the new epiBAT method achieves similar or better results than the compared methods.

Index Terms—bat algorithm, epistatic interactions, single nucleotide polymorphisms

I. INTRODUCTION

Development of DNA sequencing technologies rapidly increases the quantity of the genomic data needed to analyze. Genome-wide association studies (GWAS) focus on detecting associations between traits, mainly diseases, and genetic variants, such as single nucleotide polymorphisms (SNPs) [1]. However, to detect associations with complex diseases it is needed to consider not just one SNP, but k -way epistatic interactions, i.e. the case when k SNPs interact with each other. This greatly increases computational burden, as the number of possible interacting combinations grows exponentially with k and the quantity of SNPs.

In this paper, we introduce epiBAT, a new algorithm based on bat algorithm [2] with multiple objective functions. The structure of the paper is as follows. At first, related work is briefly described and the general bat algorithm is introduced. Then our method using bat algorithm for detecting epistatic interactions is presented.

We will use the following notation in the text. GWAS datasets represent case-control datasets with n SNPs. Possible genotype values at SNP locuses, are 0,1,2, which represent major homozygous, heterozygous and minor homozygous alleles respectively. The parameter k will represent the epistatic order, i.e. the number of interacting SNPs that are being considered for an association with a disease.

A. Related Work

There are various approaches for detecting epistatic interactions. The most straightforward approach is the exhaustive search, which evaluates every possible k -combinations of SNPs. The application of this approach is not very feasible, whereas the number of combinations grows exponentially with k and n , i.e. $\binom{n}{k}$.

A well-known method which focuses to improve the exhaustive search is Boolean Operation based Screening and Testing (BOOST) [3]. BOOST first filters SNP combinations by likelihood ratio and then evaluates left ones. BOOST also uses boolean data representation to decrease space complexity and improve CPU efficiency. Another algorithm - BEAM uses Markov Chain Monte Carlo sampling [4] to reduce the number of SNP combinations needed to evaluate.

Other approaches, using statistical methods, or filtering methods based on biological knowledge, are concisely described in [5], [6]. Some approaches are based on machine learning, such as BNMBL [7] which is based on Bayesian networks (BN), that are used to model the association between SNP combination and a disease. BN are then evaluated by different scoring functions, such as K2 score, to find how well is a disease modelled by SNP combination.

Another approaches are swarm-based algorithms or heuristic search algorithms. As objectives, they use scoring functions from machine learning, regression, or statistic tests. One of the first examples of this approach is AntEpiSeeker [8] which uses ant colony optimization (ACO) and χ^2 test statistics to score SNP combinations. ACO is also used in MACOED [9], which uses two different scores: K2 score used in BN, and Akaike Information Criterion (AIC) score, used in logistic regression. Another multi-objective algorithm based on ACO is FAACOSE [10], which uses AIC score and explain score. ACO-Tabu is also based on ACO and uses tabu search [11]. Many other algorithms for epistasis detection based on ACO exist, and they are reviewed in [12].

Other algorithms based on swarm-based algorithms are CSE [13], which uses cuckoo search algorithm, IOBLPSO [14] based on particle swarm optimization, FHSA-SED [15] and NHSA-DHSC [16] based on harmony search algorithm. FHSA-SED and NHSA-DHSC use Gini score and K2 score,

while NNSA-DHSC uses also the third objective - joint entropy. IOBPSO is based on mutual information score, while CSE uses the Bayesian score.

B. General Bat algorithm

Bat algorithm is a swarm intelligence algorithm drawing inspiration from the echolocation of bats [2]. When describing the bat algorithm, m denotes the population size, T denotes the number of maximum iterations, t denotes the t -th iteration, and i denotes the i -th agent of a population, where $i \in \{1, \dots, m\}$ and $t \in \{1, \dots, T\}$.

The i -th bat is located at a position x_i and has the following properties: velocity v_i , loudness A_i , pulse emission rate r_i and pulse frequency f_i . The pulse frequency f_i is bounded by two user input parameters, $freq_{min}$ and $freq_{max}$, and a vector β of k dimensions taking random values in each iteration from $[0, 1)$ in the following manner:

$$f_i = freq_{min} + (freq_{max} - freq_{min})\beta. \quad (1)$$

The general bat algorithm consists of three steps executed in each iteration: flight of the bats, the local search, and acceptance of new solutions. In the first step, the bat with the best score b_{best} is found, then the velocity v_i and position x_i of the i -th bat for $i \in \{1, \dots, m\}$ is updated as follows:

$$v_i = v_i + (x_i - x_{best})f_i, \quad (2)$$

$$x_i = x_i + v_i. \quad (3)$$

The local search step is controlled by the r_i (pulse emission rate) and A_i (loudness) parameters. The parameter r_i of a bat controls if the bat should do a local search, while the average loudness of all bats in a population A_{avg} at t -th iteration, bounds the area of local random search. Local search for the i -th bat is then defined as:

$$x_i = x_i + \epsilon, \quad (4)$$

where ϵ is drawn from $[-A_{avg}, A_{avg}]$. As iterations proceed and bats move to better solutions, the area of local search decreases in size with the A_i decreasing, while the chance for a bat to do the local search increases:

$$A_i = A_i\alpha, \quad (5)$$

$$r_i = r_i[1 - \exp(-\gamma t)], \quad (6)$$

where α and γ are constant and user defined parameters, while usually these constraints hold: $0 < \alpha < 1$ and $\gamma > 0$.

In the last step, the new positions of the bats are evaluated for acceptance, which is for i -th bat controlled by the A_i parameter. New position is accepted only when its score has improved compared to the previous iteration. The overview of the general bat algorithm can be summarized as:

```

while  $t < T$  do
  for  $i = 1$  to  $m$  do
    adjust  $f_i$ , update  $v_i$  and  $x_i$ 
    if  $rand > r_i$  then
      do local search
  
```

```

    end if
    if  $rand < A_i$  and  $b_i$  is better than  $b_i$  at  $t - 1$  then
      accept the new solution  $x_i$ 
      update  $r_i$  and  $A_i$ 
    end if
  end for
end while
  
```

II. DESCRIPTION OF THE EPIBAT ALGORITHM

The EpiBAT algorithm consists of two main stages. The first stage finds the potential SNP combinations i.e. the candidate set, while in the second stage SNP combinations are evaluated by statistical test for significance. To obtain the candidate set, epiBAT uses modified bat algorithm and the tabu search.

Bats in the epiBAT algorithm represents coordinates in the continuous searching space of k dimensions, where the space is bounded by n in each dimension. To score the bats as SNP combinations, they are rounded to the nearest integers. For coordinates of the bats, rule of ascending order of the coordinates is always enforced. Whenever a bat moves out of the searching space in one dimension, its coordinate is clipped to the nearest boundary value in that dimension, and its velocity is also reinitialized in that dimension. Also as bats should always represent k -way SNP combination, whenever this fact does not hold (i.e. bat's coordinates are the same for two dimensions after rounding), the position of that bat is changed by 1 in a random dimension.

In the epiBAT algorithm, the local search step of the general bat algorithm is modified by adjusting ϵ value in (4), so that ϵ is generated in one random dimension from $[0, n]$. This allows the bat to search all SNPs in one dimension. As better solutions are found, loudness decreases and searching space in other dimensions is less than 1, thus bats search only in one dimension, to find the best SNP combination of that SNP.

A. Objective functions

Our algorithm uses two objective functions: Gini score and K2 score. To compute both scoring functions, contingency tables are used, which display frequency distribution of two variables I and J , where I represents a number of all genotype combinations of k SNPs, and J represents a phenotype state (i.e. case or control). For a k -way SNP combination, $I = 3^k$, as SNP has 3 possible genotype values, and $J = 2$, as there are cases or controls. The i -th genotype combination has n_{ij} samples with phenotype j , and n_i samples overall. Then the K2 score is defined as follows:

$$K2_score = \prod_{i=1}^I \left(\frac{(J-1)!}{(n_i + J + 1)!} \prod_{j=1}^J n_{ij}! \right). \quad (7)$$

Whereas factorials can produce values that cannot be stored in computer memory, epiBAT uses logarithmic version of K2 score [9]:

$$K2_score = \sum_{i=1}^I \left(\sum_{b=1}^{n_i+1} \log(b) - \sum_{j=1}^J \sum_{d=1}^{n_{i,j}+1} \log(d) \right). \quad (8)$$

The Gini score is defined as follows:

$$Gini_score = \sum_{i=1}^I P_i \left(1 - \sum_{j=1}^J p_{ij}^2\right), \quad (9)$$

where P_i is a percentage of i -th genotype combination in a whole dataset, and p_{ij} represents the percentage of samples with i -th genotype combination and phenotype j , i.e. $p_{ij} = \frac{n_{ij}}{n_i}$.

In the epiBAT algorithm, two separate populations of bats are used, whereas one population has the Gini score as the objective function, and the second population has the K2 score. This approach of separate populations is used in [16], and it differs from most of the other methods, which uses Pareto optimal approach, such as MACOED [9]. In each iteration, the best bats of both populations are stored in the candidate set. Also to prevent bats staking at some local optima, the tabu search is used, whereas each population has its own tabu table.

B. Tabu search

When the best solution of one population has not been improved for Q iterations, where Q is a user defined parameter, the best solution is added to the tabu table for that population. Then, all possible combinations of the bats with one of the SNPs of the best solution are generated and scored. All combinations with scores approximately better than the best solution is added to the candidate set. The approximate comparison is controlled by the δ parameter, i.e. the combination with score s is added to the candidate set, when $s * \delta$ is better than the score of the best solution.

Then all bats of that population that are located at one locus of the best bat, are removed from that population and new bats are generated. Also bats are checked in each iteration if they did not move to the tabu position, i.e. when one coordinate of a bat is located in the tabu table of the corresponding population. Otherwise, they are removed from that population and new bats are created. This forces bats to search unexplored areas of space, and to search for SNP combinations with little marginal effects.

C. Evaluation of the candidate set

To evaluate the SNP combinations from the candidate set, various statistical tests are used, for example χ^2 test in MACOED [9], or Fisher test in FAACOSE [10]. Then only those SNP combinations are reported, which are passing the significance level α , which is adjusted by Bonferroni correction: $\alpha_{bonf} = \frac{\alpha}{\binom{k}{i}}$. In the epiBAT algorithm G -test [17] is used, which is more recent statistical significance test, which performs better than χ^2 test in most cases [16]. The G -test is defined as follows:

$$G = 2 \sum_{i=1}^I \sum_{j=1}^J n_{ij} \ln\left(\frac{n_{ij}}{E_{ij}}\right), \quad (10)$$

where n_{ij} is the number of samples with i -th genotype combination and j -th phenotype state (i.e. case or control),

and E_{ij} is the expected number of samples with i -th genotype combination and j -th phenotype state.

However, it is known that these tests do not give the best results when a cell in a contingency table contains less than 5 samples. Therefore, some methods like [16], [18] modify the statistical test. In our method, we remove the columns of the contingency table that contain less samples than ζ , and also correct the number of degrees of freedom accordingly. Experimentally, we set $\zeta = C/100$, where C is the total number of cases and controls.

To minimize the quantity of false positives, the epiBAT algorithm uses a similar technique as AntEpiAco [19], i.e. SNP combination S is reported only when its p-value is lower than p-values of all the other SNP combinations, which shares at least one SNP with the S . This reduces the number of outputted SNP combinations in cases, where many SNP combinations of one SNP can pass the significance level even after Bonferroni correction.

To summarize, the overview of the epiBAT algorithm is as follows:

- 1) Initialize bat populations.
- 2) Find the best bat for each population and store them in the candidate set.
- 3) Move bats.
- 4) Execute modified local search.
- 5) Tabu search
- 6) Replace bats located in tabu table with new ones.
- 7) Repeat steps 2-6 for the defined number of iterations.
- 8) Evaluate the candidate set by the modified G -test.

III. EXPERIMENTS

The epiBAT algorithm is written in Python with usage of Numpy¹ and Pandas² library, and its implementation is available at <https://github.com/xsitarcik/epiBAT>. In this section, the performance of the epiBAT algorithm is compared to other methods, including MACOED, BOOST, BEAM and AntEpiSeeker.

A. Datasets

For comparison with other methods we use the same testing datasets as used in MACOED [9]. MACOED used three epistasis models with marginal effects, where for each epistatic model, the parameter minor allele frequency (MAF) varied with four values (0.05, 0.1, 0.2 and 0.5), thus 12 penetrance tables were obtained in total (denoted as DME1_1,...,DME1_4, DME2_1,...,DME2_4, DME3_1,...,DME3_4) and 100 datasets were generated for each penetrance table. In each dataset there are 1600 SNPs and only one 2-combination of SNPs is correct.

SNP locuses were randomly reshuffled for epiBAT, as the performance of the bat algorithm depends how the SNPs are positioned in the searching space.

¹<https://numpy.org/>

²<https://pandas.pydata.org/>

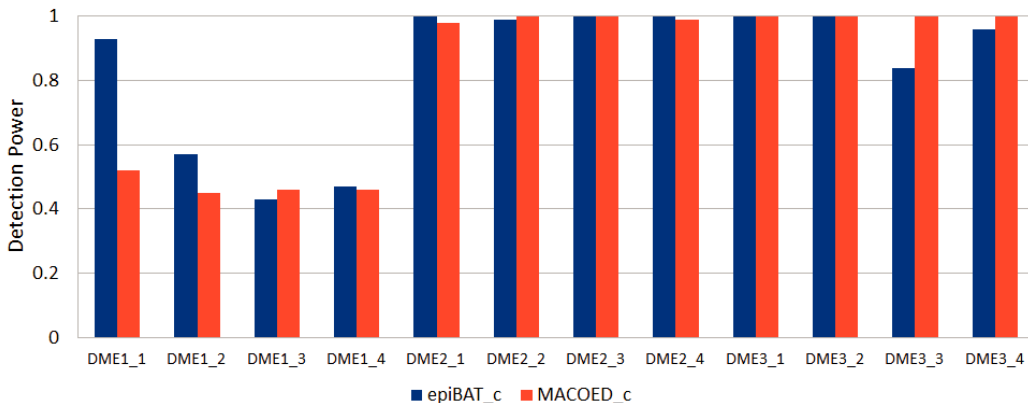


Fig. 1. Detection power of MACOED’s and epiBAT’s candidate sets on datasets generated from different penetrance tables.

B. Performance evaluation

Detection power D is used to evaluate performance, and is described as follows:

$$D = \frac{S_{correct}}{S_{all}}, \tag{11}$$

where $S_{correct}$ is the quantity of datasets where correct solution was reported, and S_{all} denotes the quantity of all tested datasets. In our experiments, each dataset was tested 5 times, then the result was averaged.

We have also used other metrics: recall (R), precision (P) and F-measure (F), as it was suggested in MACOED [9]. These metrics are based on confusion matrix consisting of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). TP represent the quantity of correctly reported SNP combinations, FN represent the quantity of correct SNP combinations that were not reported, and FP indicates the quantity of falsely reported SNP combinations. Then these metrics are defined as follows:

$$\begin{aligned} R &= \frac{TP}{TP + FN}, \\ P &= \frac{TP}{TP + FP}, \\ F &= \frac{1}{2P + 2R}. \end{aligned} \tag{12}$$

We measured the detection power for the final results of epiBAT, but also the detection power of the candidate set of epiBAT, we denote these results in figures as epiBAT_c, and MACOED’s candidate set is denoted as MACOED_c. Recall, precision and F-measure, were measured only for the final stage of epiBAT.

C. Parameter settings

The following parameter settings of the epiBAT algorithm were used for each dataset. The loudness of the bats A_i was initialized randomly from uniform distribution $[2, 4)$, pulse emission rate of the bats r_i was initialized from $[0, 0.5)$, and velocity v_i from $[0, 1)$. As used in [2], we set $\gamma = 0.9$ and

$\alpha = 0.7$. The number of iterations Q that are required to add the best solution into tabu region was set to 7.

Usually, for a dataset with 100 SNPs, the total number of iterations is 50, and the number of agents is 50 as well. In our case, we used the double number of agents as we use two populations of agents. Then, to make a fair comparison, we halved the number of iterations. So the total number of iterations was 25, and the population size was 100; 50 for population with K2 score as objective function and 50 for population with Gini score.

For $freq_{min}$ and $freq_{max}$ we used 0.05 and 0.2 respectively. The parameter ζ used in approximate comparison was set to 1.015 for both scores. The parameter β was drawn randomly from uniform distribution $[0, 1]$. We used $\alpha = 0.1$, the same as in the MACOED experiments.

D. Results

The results of the compared methods are taken from the paper introducing MACOED [9].

Fig. 1 shows the results of detection power of MACOED’s and epiBAT’s candidate sets. Both MACOED and epiBAT use K2 score as objective function, but they differs in the second objective, whereas MACOED uses AIC (Akaike Information Criterion) score and epiBAT uses Gini score. In MACOED, the candidate set consists of all nondominated solutions from the reported solutions. As epiBAT’s candidate set is larger, we selected only the best 5 solutions per objective function, to make a fairer comparison. EpiBAT achieved significantly higher detection power in DME1_1 and DME1_2, while significantly lower detection power was in DME3_3. In other models, results were comparable. These results confirm the findings of [15], that Gini score performs well mainly on models with low genetic heritability like DME1_1 model, where epiBAT_c performed overwhelmingly better than MACOED_c which does not use Gini score. On the other hand, lower detection of EpiBAT in DME3_3 may be caused by not using AIC score.

Fig. 2 and Fig. 3 show the results of detection power of the final stages of MACOED and epiBAT (i.e. after the evaluation of the candidate set by the statistic test), and other methods.

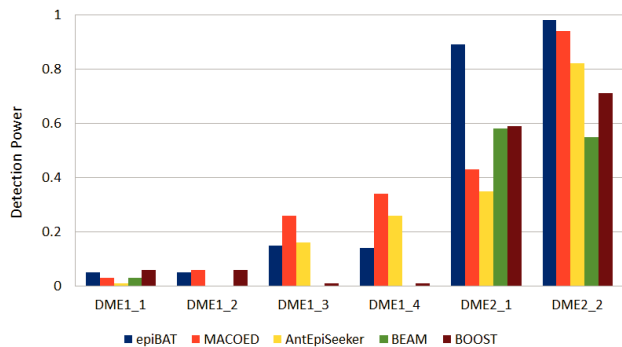


Fig. 2. Comparisons of detection power on the first six DME models.

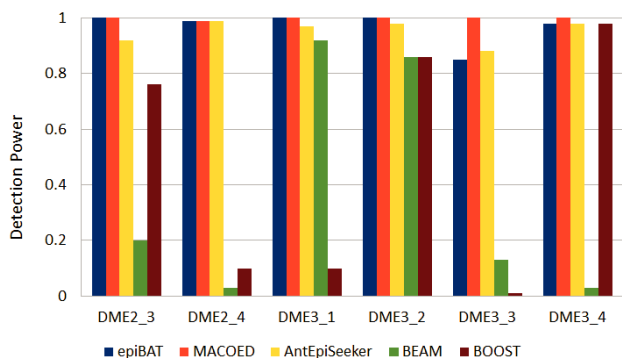


Fig. 3. Comparisons of detection power on the second six DME models.

EpiBAT achieved the highest detection power in DME2_1 and DME2_2, while it performed worse in DME1_3, DME1_4, and DME3_3. Although the detection power of epiBAT_c in DME1_1 was very high, only a few passed the G -test, thus the final detection power was very low, but still comparable to others.

Table I, Table II, and Table III show the results for recall, precision and F-measure metrics. Although the G -test is more recent and more preferred statistic test for evaluating the epistasis significance, it did not perform that well in these datasets as compared to the χ^2 test with number of degrees of freedom fixed to eight, which was used in MACOED. In epiBAT, the number of degrees of freedom was not fixed, but dependent on the number of rows and cells in the contingency table of the SNP combination. This could lead to the situation, where correct SNP combination did not passed the test, or other SNP combination sharing the same locus as the correct combination had higher p-value.

IV. CONCLUSION

In this paper, we have presented a new algorithm for detecting epistasis interactions called epiBAT, which is based on the bat algorithm and tabu search. To the best of our knowledge, we are the first to implement bat algorithm for epistasis detection. EpiBAT uses Gini score and K2 score as

TABLE I
RECALL, PRECISION AND F-MEASURE ON THE FIRST DME MODEL

Model	Method	Recall ^a	Precision ^a	F-measure ^a
DME1_1	epiBAT	0.05	0.45	0.09
	MACOED	0.03	0.43	0.06
	AntEpiSeeker	0.01	0.25	0.02
	BEAM	0.03	0.19	0.05
	BOOST	0.06	0.1	0.07
DME1_2	epiBAT	0.05	0.45	0.09
	MACOED	0.06	0.86	0.11
	AntEpiSeeker	0	0	0
	BEAM	0	0	0
	BOOST	0.06	0.11	0.08
DME1_3	epiBAT	0.17	0.43	0.25
	MACOED	0.26	0.74	0.39
	AntEpiSeeker	0.16	0.7	0.26
	BEAM	0	0	0
	BOOST	0.01	0.01	0.01
DME1_4	epiBAT	0.22	0.35	0.27
	MACOED	0.34	0.45	0.39
	AntEpiSeeker	0.26	0.67	0.37
	BEAM	0	0	0
	BOOST	0.01	0.02	0.01

^aThe best result is in bold.

TABLE II
RECALL, PRECISION AND F-MEASURE ON THE SECOND DME MODEL

Model	Method	Recall ^a	Precision ^a	F-measure ^a
DME2_1	epiBAT	0.88	0.92	0.9
	MACOED	0.43	0.98	0.6
	AntEpiSeeker	0.35	0.92	0.51
	BEAM	0.58	0.72	0.64
	BOOST	0.59	0.51	0.55
DME2_2	epiBAT	0.98	0.97	0.98
	MACOED	0.94	1	0.97
	AntEpiSeeker	0.82	0.91	0.86
	BEAM	0.55	0.48	0.51
	BOOST	0.71	0.56	0.63
DME2_3	epiBAT	1	1	1
	MACOED	1	0.96	0.98
	AntEpiSeeker	0.92	0.94	0.93
	BEAM	0.2	0.12	0.15
	BOOST	0.76	0.51	0.61
DME2_4	epiBAT	0.99	0.99	0.99
	MACOED	0.99	0.94	0.97
	AntEpiSeeker	0.99	0.98	0.99
	BEAM	0.03	0.01	0.02
	BOOST	0.1	0.12	0.11

^aThe best result is in bold.

objective functions, and two populations of agents, each with the different objective function.

Although the proposed algorithm performs well and comparably to other methods, further research is needed to achieve the most optimal parameter settings of the epiBAT algorithm. The other future work can consist of experimenting with various statistical tests used for evaluating the candidate set. This is an important area, as it can generate many false positives, or only a few candidates can pass the test. In our future work we also want to apply epiBAT for epistasis detection of higher epistatic order.

TABLE III
RECALL, PRECISION AND F-MEASURE ON THE THIRD DME MODEL

Model	Method	Recall ^a	Precision ^a	F-measure ^a
DME3_1	epiBAT	1	1	1
	MACOED	1	1	1
	AntEpiSeeker	0.97	0.96	0.97
	BEAM	0.92	0.77	0.84
	BOOST	0.1	0.12	0.11
DME3_2	epiBAT	1	1	1
	MACOED	1	1	1
	AntEpiSeeker	0.98	0.99	0.98
	BEAM	0.86	0.75	0.8
	BOOST	0.86	0.57	0.69
DME3_3	epiBAT	0.87	0.99	0.92
	MACOED	1	1	1
	AntEpiSeeker	0.88	0.99	0.93
	BEAM	0.13	0.32	0.18
	BOOST	1	0.63	0.77
DME3_4	epiBAT	0.98	0.96	0.97
	MACOED	1	0.99	1
	AntEpiSeeker	0.98	0.96	0.97
	BEAM	0.03	0.02	0.03
	BOOST	0.98	0.65	0.78

^aThe best result is in bold.

ACKNOWLEDGMENT

This work was partially supported by the Scientific Grant Agency of The Slovak Republic, Grant No. VG 1/0458/18, and APVV-16-0484.

REFERENCES

[1] J. N. Hirschorn and M. J. Daly, “Genome-wide association studies for common diseases and complex traits,” in *Nat Rev Genet.*, vol. 6, no. 2, pp. 95-108, 2005.

[2] X. S. Yang, “A new metaheuristic bat-inspired algorithm,” in: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. *Studies in Computational Intelligence*, vol. 284, Springer, Berlin, Heidelberg, 2015.

[3] X. Wan, et al., “BOOST: A Fast Approach to Detecting Gene-Gene Interactions in Genome-wide Case-Control Studies,” in *American journal of human genetics*, vol. 87, no.3, pp. 325-340, 2010.

[4] Y. Zhang and J. S. Liu, “Bayesian inference of epistatic interactions in casecontrol studies,” in *Nature genetics*, vol. 39, no. 9, pp. 1167-1173, 2007.

[5] X. Sun, et al., “Analysis pipeline for the epistasis search - statistical versus biological filtering,” in *Frontiers in Genetics*, vol. 5, no. 106, 2014.

[6] C. Niel, C. Sinoquet, C. Dina, and G. Rocheleau, “A survey about methods dedicated to epistasis detection,” in *Frontiers in Genetics*, vol. 6, no. 285, 2015.

[7] X. Jiang, et al., “Learning genetic epistasis using Bayesian network scoring criteria,” in *BMC Bioinformatics*, vol. 12, no. 89, 2011.

[8] Y. Wang, et al., “AntEpiSeeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm,” in *BMC research notes*, vol. 3, no. 117, 2010.

[9] P.-J. Jing and H.-B. Shen, “MACOED: A multi-objective ant colony optimization algorithm for SNP epistasis detection in genome-wide association studies,” in *Bioinformatics*, vol. 31, no. 5, pp. 634-641, 2015.

[10] L. Yuan, C.-A. Yuan, and D.-S. Huang, “FAACOSE: A fast adaptive ant colony optimization algorithm for detecting SNP epistasis,” in *Complexity*, vol. 2017, 2017.

[11] E. Sapin, E. Keedwell, and T. Frayling, “An Ant Colony Optimization and Tabu List Approach to the Detection of Gene-Gene Interactions in Genome-Wide Association Studies,” in *IEEE Computing Intelligence Magazine*, vol. 10, no. 4, pp. 54-65, November 2015.

[12] J. Shang, et al., “A review of ant colony optimization based methods for detecting epistatic interactions,” in *IEEE Access*, vol. 7, pp. 13497–13509, 2019.

[13] M. Aflakparast, et al., “Cuckoo search epistasis: a new method for exploring significant genetic interactions,” in *Heredity*, vol. 112, no. 6, pp. 666–674, 2014.

[14] J. Shang, et al., “An improved opposition-based learning particle swarm optimization for the detection of SNP-SNP interactions,” in *BioMed Research International*, vol. 2015, 2015.

[15] S. Tuo, J. Zhang, X. Yuan, Y. Zhang, and Z. Liu, “FHSA-SED: Two-Locus Model Detection for Genome-Wide Association Study with Harmony Search Algorithm,” in *PLoS one*, vol. 11, no. 3, 2016.

[16] S. Tuo, J. Zhang, X. Yuan, Z. He, Y. Liu, and Z. Liu, “Niche harmony search algorithm for detecting complex disease associated high-order SNP combinations,” in *Scientific Reports*, vol. 7, no. 1, December 2017.

[17] J. H. McDonald, *G-Test of goodness-of-fit*. *Handbook of Biological Statistics*, 3rd ed., Sparky House Publishing: Baltimore, MD, USA, 2014, pp. 53-58.

[18] S. Tuo, “FDHE-IW: A fast approach for detecting high-order epistasis in genome-wide case-control studies,” in *Genes*, vol. 9, no. 9, 2018.

[19] Y. Sun, J. Shang, J. Liu, S. Li, and C. Zheng, “epiACO - a method for identifying epistasis based on ant Colony optimization algorithm,” in *BioData Mining*, vol. 10, July 2017.

Novel RBF Approximation Method Based on Geometrical Properties for Signal Processing with a New RBF Function: Experimental Comparison

Vaclav Skala
 University of West Bohemia
 Pilsen, Czech Republic
<http://www.VaclavSkala.eu>

Martin Cervenka
 University of West Bohemia
 Pilsen, Czech Republic
cervemar@kiv.zcu.cz¹

Abstract— Interpolation and approximation methods are widely used in many areas. They can be divided to methods based on meshing (tessellation) of the data domain and to meshless (meshfree) methods, which do not require the domain tessellation of scattered data. Scattered n -dimensional data radial basis function (RBF) interpolation and approximation leads to a solution of linear system of equations.

This contribution presents a new approach to the RBF approximation based on analysis of geometrical properties of signals, i.e. sampled curves. Also a newly developed radial basis function was used and proved better precision of approximation.

Experimental comparison of several RBF functions (Gauss, Thin-Plate Spline, CS-RBF and a new proposed RBF) is described with analysis of their properties. Special attention was taken to the precision of approximation and conditionality issues. The proposed approach can be extended to a higher dimensional case and for vector data, e.g. fluid flow, too.

Index Terms—Approximation, Radial basis functions, RBF, Signal processing, Computer graphics, Meshless methods.

I. INTRODUCTION

Interpolation and approximation of scattered data is required in many areas. As there is no ordering defined for d -dimensional case, if $d \geq 1$, usually two approaches are taken:

- Tessellation of the data domain, e.g. using Delaunay triangulation and application of a selected interpolation or approximation method. However, the Delaunay tessellation has a computational complexity $O(n^{\lceil d^2/2 \rceil})$.

This leads high computational complexity and to implementation problems in the case that $d > 2$. Another computational problems can be expected, if smoothness of the interpolation or approximation is required.

- Use of meshless methods based on Radial Basis Functions (RBF) use leads to a solution of a linear system of equations $\mathbf{Ax} = \mathbf{b}$, in general, and the computational complexity is nearly independent from the dimensionality of the data domain, see Hardy [1], . Even more, if relevant RBF function is selected, higher degree of smoothness is obtained. On the other hand, interpolation

and approximation methods based on meshless approach usually have a problem with a precision on borders or on discontinuities, in general. The meshless methods can be also used for approximation of vector data, i.e. fluid flow etc. However, some RBFs applications might lead to numerical problems due to ill-conditioned matrix of the linear system, especially for large data sets.

Usually, the approximation methods use a general method, which is not taking directly geometrical properties of the signal into account, e.g. Singh [2], Skala & Smolik & Nedved [3]. However, if some information on signal geometry can be extracted from data and used, the approximation should be more precise and simpler. Such approach has been used by Majdisova & Skala & Smolik [4]. This approach seems to be quite complicated, as it is based on properties of cubic curve in floating data window.

This contribution is focused on the following main aspects:

- how geometrical properties of a signal can be efficiently used for good and robust approximation,
- how to approximate a signal, i.e. a sampled curves, with a good precision with a minimal number of radial basis functions (RBFs),
- what kind of RBFs probably gives better results,
- what are properties of a newly developed RBF in terms of precision and numerical precision.

As some RBF functions have a parameter, called a shape parameter, some problems can be expected with an optimal shape parameter selection or estimation. Some proposal how to select suitable shape parameters were introduced by Karageorghis [5], Wang & Liu [6] for range of shape parameters generation, Afiatdoust & Esmaeilbeigi [7] presented use of genetic algorithm. Moreover, Sarra & Sturgill [8] propose non-deterministic approach based on random shape parameter generation. However, some experiments recently made by Skala & Karim & Zabran [9] proved, that there is probably no optimal constant shape parameter nor one vector of optimal shape parameters for each RBF.

Another problem is shape parameter selection Karageorghis [5]. There are two particular cases which may occur in

The research was supported by Czech Science Foundation (GACR) project No.GA 17-05534S and partially by SGS 2019-016 project

¹ Corresponding author

general: approximation will be inaccurate or the problem may become ill-conditioned. That is why correct shape parameter selection is needed. Some approaches to select suitable shape parameters already exists. Approach introduced by Wang & Liu [6] for example generates range of shape parameters, Afiatdoust & Esmailbeigi [7] presented theirs approach using genetic algorithm. Moreover, Sarra & Sturgill [8] propose non-deterministic approach based on random generator. Some recent research have been devoted to variable shape parameters, i.e, each RBF function has a different shape parameter, e.g. Majdisova [4], Skala [9].

Application of the RBF interpolation and approximation in engineering practice can be found in Biancolini [10], Fasshauer [11], Menandro [12]. Also RBFs are used for vector field interpolation and approximation, e.g. Smolik [13], [14], [15], [16], Skala [17], solution of partial differential equations (PDE) Zhang [18], Neural Networks RBF Yinwey [19] and reconstruction of implicit curves Cuomo [20]. Comparison of selected RBFs can be found in Majdisova [21].

II. RBF INTERPOLATION

According to Hardy [22], RBF interpolation is based on determining the distance of two point (in the d -dimensional space in general). The interpolation is given in the form:

$$h(\mathbf{x}) = \sum_{j=1}^N \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) = \sum_{j=1}^N \lambda_j \varphi(r_j) \quad (1)$$

where r_i is the distance from a point x to the point x_i . As the parameter of the function φ is a distance of two points in the d -dimensional space, the interpolation is non-separable by a dimension. The RFBs will be described in detailed latter on.

For each point x_i the interpolating function has to have value h_i . Therefore, we are getting a system of linear equations:

$$h(\mathbf{x}_i) = \sum_{j=1}^N \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) = \sum_{j=1}^N \lambda_j \varphi(r_{ij}) \quad (2)$$

where λ_j are unknown weights for each radial basis function, N is the number of given points and φ is the radial basis function itself. It can be rewritten if the matrix form as:

$$\mathbf{A}\lambda = \mathbf{h} \quad (3)$$

or in a detailed form as (4):

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1j} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi_{i1} & \cdots & \varphi_{ij} & \cdots & \varphi_{iN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{Nj} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{bmatrix} \quad (4)$$

After solving the system of linear equations, interpolated value at the point \mathbf{x} is computed using (1). However, due to numerical robustness and stability, additional polynomial

conditions are usually added Skala [23] [24]. In the case of an additional polynomial we obtain:

$$h(\mathbf{x}_i) = \sum_{j=1}^N \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) + P_k(\mathbf{x}_i) \quad (5)$$

In the case of a linear polynomial (in $2\frac{1}{2}$ D case):

$$P_k(x, y) = a_0 + a_1x + a_2y \quad (6)$$

This additional conditions can be rewritten as:

$$\sum_{j=1}^N \lambda_j = 0 \quad \sum_{j=1}^N \lambda_j \mathbf{x}_j = 0 \quad (7)$$

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1N} & 1 & x_1 & y_1 \\ \vdots & \ddots & \vdots & 1 & \vdots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{NN} & 1 & x_N & y_N \\ 1 & 1 & 1 & 0 & 0 & 0 \\ x_1 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & \cdots & y_N & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

This matrix can be further rewritten in more compact way:

$$\begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix} \quad (9)$$

The matrix \mathbf{P} represents polynomial additional conditions, λ is vector of RBF weights, vector \mathbf{a} contains resulting polynomial coefficients and \mathbf{h} are given values at the given points.

It should be noted that in some cases that it can be counterproductive especially for large scope of domain data Jäger [25], Skala [23] [24].

III. RBF APPROXIMATION

Approximation methods are slightly different from interpolation as the final approximated curve does not need to "pass" all the given points. If the matrix \mathbf{A} is a square matrix (RBF count M is equal to size of the \mathbf{x} vector), this is an interpolation problem. On the other hand, when M is smaller, it becomes approximation problem, because equation system became over-determined. Let us ξ_j are significant points in the given signal, then the approximation can be formulated as:

$$h(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \xi_j\|) \quad (10)$$

where λ_j are weights of each radial basis function, M is count of RBF being used and $M \ll N$, the φ is the RBF, ξ_j are center (important) points. Then the approximation can be expressed by equation (11):

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1M} \\ \vdots & \ddots & \vdots \\ \varphi_{i1} & \cdots & \varphi_{iM} \\ \vdots & \ddots & \vdots \\ \varphi_{N1} & \cdots & \varphi_{NM} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_i \\ \vdots \\ h_N \end{bmatrix} \quad (11)$$

Overdetermined linear equation system is to be solved, e.q. by the Least Square Error method (12).

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{h} \quad (12)$$

However, in the approximation case, the additional polynomial conditions cannot be used Majdisova [26], [27].

IV. RBF FUNCTIONS

There are several radial basis functions Fasshauer [11], Majdisova [28] [29], Buhmann [30]. They can be divided into two major groups (Smolik [13]):

- "global" RBFs having global influence, e.g. $r^2 \lg(r)$, $\exp(-\alpha r^2)$, $\sqrt{\alpha + r^2}$, $1/\sqrt{\alpha + r^2}$, $1/(\alpha + r^2)$, etc. where α is a shape parameter. The RBF matrix is usually full and ill conditioned.
- "local" RBFs - Compactly Supported RBF (CS-RBF) have a non-zero value for the interval $\langle 0, 1 \rangle$ only. The RBF matrix is usually sparse as it depends on the scaling of the interval $\langle 0, 1 \rangle$ to the required one. Some examples are listed in Tab.I and they are shown on Fig.1. The article Menandro [12] describes this class of RBFs.

Some examples of RBFs are listed in Tab.I and at Fig.1.

ID	Function
1	$(1-r)_+$
2	$(1-r)_+^3 (3r+1)$
3	$(1-r)_+^5 (8r^2+5r+1)$
4	$(1-r)_+^2$
5	$(1-r)_+^4 (4r+1)$
6	$(1-r)_+^6 (35r^2+18r+3)$
7	$(1-r)_+^8 (32r^3+25r^2+8r+3)$
8	$(1-r)_+^3$
9	$(1-r)_+^5 (5r+1)$
10	$(1-r)_+^7 (16r^2+7r+1)$

TABLE I: List of well-known CS-RBF.

V. PROPOSED RBF

In this paper a new global radial basis function is proposed. It has one shape parameter and is defined as is in (13).

$$\varphi(r) = r^2 (r^\alpha - 1) \quad (13)$$

where α is a shape parameter (we use $\alpha = 1.8$ globally). The function is shown at Fig.2.

VI. DESCRIPTION OF EXPERIMENTS

For sake of simplicity, all signal values has been normalized to interval $h(x_i) \in \langle 0, 1 \rangle$, i. e. $y_i = h(x_i)$. Signal domain has been set to the $x_i \in \langle 0, 1 \rangle$ as well for the same reason.

As already mentioned, four RBF has been used for testing purposes. Gaussian RBF (in equation (14)) is global radial basis function with one shape parameter α defining its dispersion.

$$\varphi(r) = e^{-\alpha r^2} \quad (14)$$

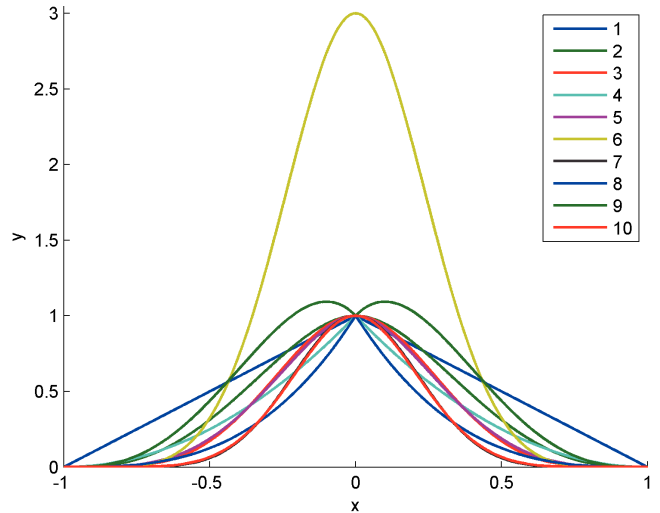


Fig. 1: Plotted CSRBFs taken from [13].

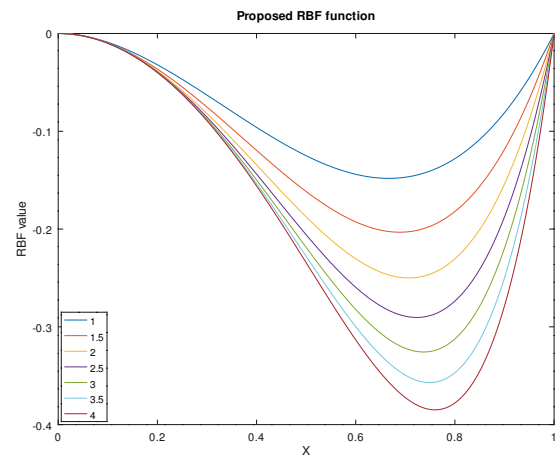


Fig. 2: Proposed RBF function with various shape parameters.

Thin plate spline (TPS) function is the next one which has been tested. It is global RBF as well, and it is defined in (15).

$$\varphi(r) = r^2 \log r \quad (15)$$

Next class of function on the list is CS-RBF. In particular, (16) function has been selected from the Tab.I. It is worth noting that this function (like all CS-RBF) is local.

$$\varphi(r) = (1-r)_+^7 (16r^2 + 7r + 1) \quad (16)$$

Last but not least we propose another RBF. It is global RBF with a shape parameter α and it is described by equation (13).

Described radial basis functions approximation has been tested against multiple signals, however, there are listed only some of tested signals in this paper. This signal subset contains following functions:

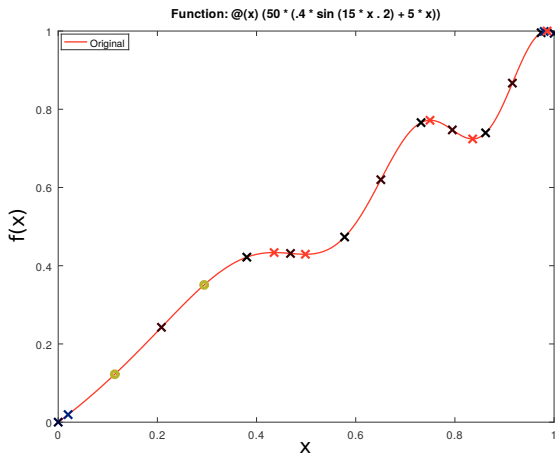


Fig. 3: Function 1.

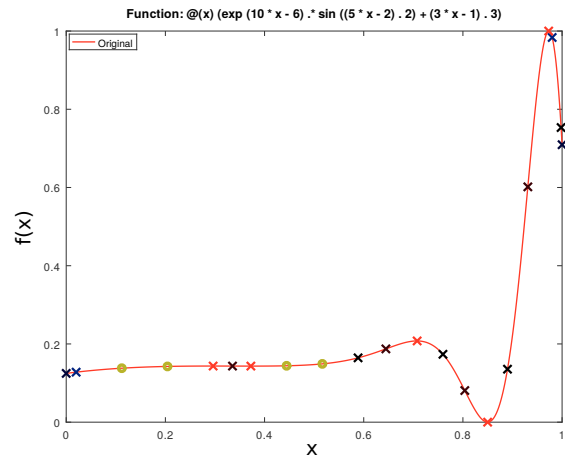


Fig. 5: Function 3.

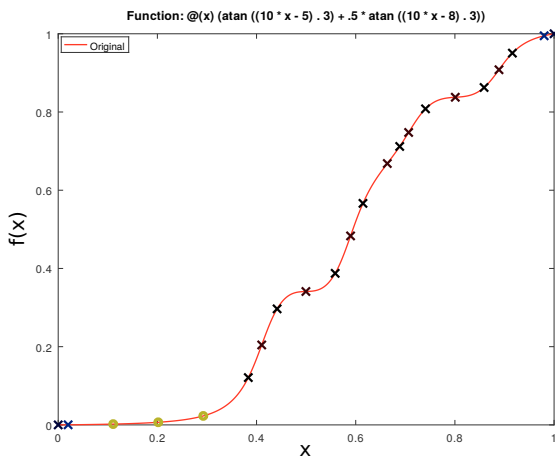


Fig. 4: Function 2.

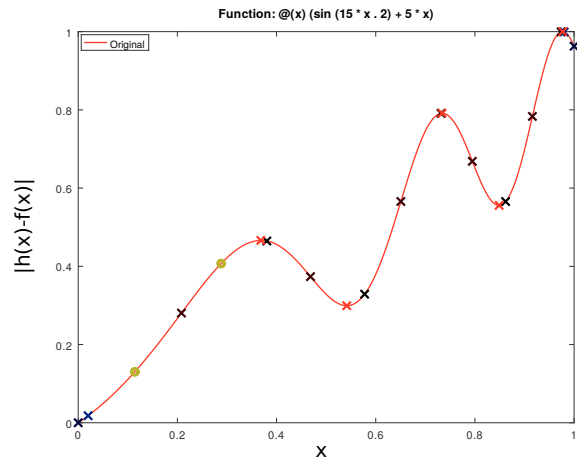


Fig. 6: Function 4.

- 1) $f_1(x) = 50(0.4\sin(15x^2) + 5x)$
- 2) $f_2(x) = \text{atan}\left((10x - 5)^3\right) + 0.5\text{atan}\left((10x - 8)^3\right)$
- 3) $f_3(x) = e^{10x-6}\sin\left((5x - 2)^2\right) + (3x - 1)^3$
- 4) $f_4(x) = \sin(15x^2) + 5x$

Selected signals (sampled curves) are shown at Fig.3 – 6.

The RBF center points ξ_i are shown as various marks on plotted signal curve on each plot. Four already mentioned RBFs (Gauss, TPS, CS-RBF and the proposed one) were selected to approximate these signals (among others).

VII. EXPERIMENTAL RESULTS

The proposed approach was tested on several testing functions, see Tab.II. It should be noted explicitly, that all function were normalized for the interval $x \in \langle 0, 1 \rangle, y \in \langle 0, 1 \rangle$. In order to easily compare errors of the proposed RBF approximation methods.

As the proposed RBF approximation is based on finding significant geometric properties, such as maxima, minima, inflection points, etc., the conditionality of the RBF metrics and

mutual comparison of errors were analyzed. Error behaviour is considered as the critical issue in approximation in general.

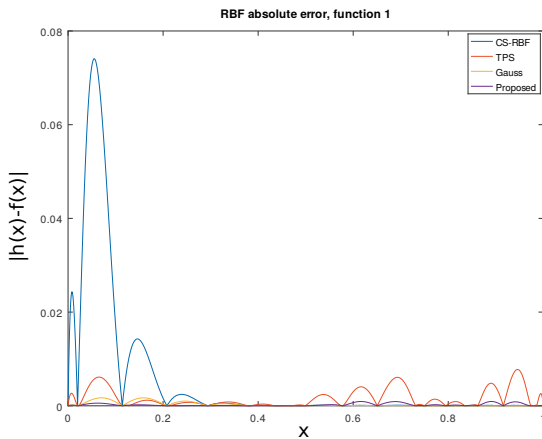
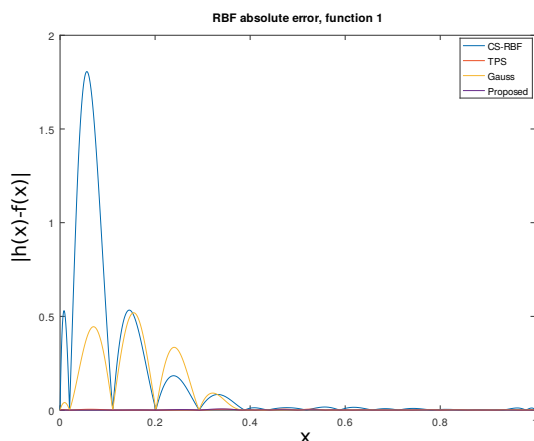
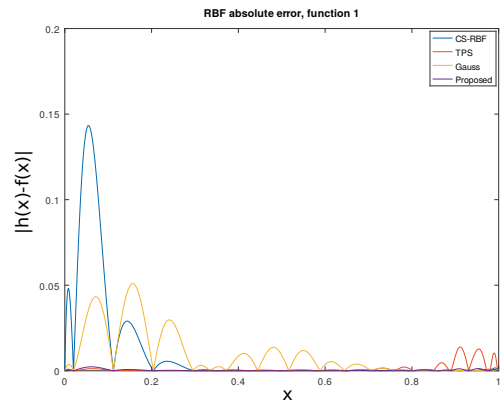
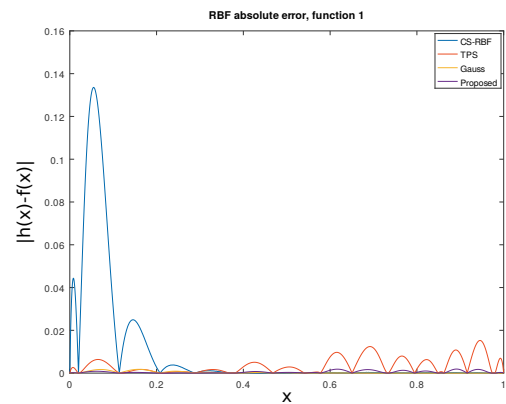
In this contribution only couple of function used are presented, see Fig.3 – 6. The relevant approximation error behaviour is presented at Fig.7 – 10.

Tab.III – V present the error behaviour numerically. Exact numerical experimental results are presented in following Tab.III (mean square error), Tab.IV (maximum absolute error) and Tab.V (condition numbers of equation system matrix \mathbf{A} defined in equation 3) respectively. It can be seen that the high error is caused by significant under-sampling. Inclusion of additional point(s) leads to significant decrease of the approximation error. It should be noted, that this contribution is analyzing the approximation behaviour at the lowest border of the sampling frequency.

The experiments proved that the sampled curves can be efficiently approximated by the few "important" points, i.e. extrema, inflections etc., with acceptably low error. The proposed method also leads to good data compression.

1	$\sin(15x^2) + 5x$
2	$0.5 \cos(20x) + 5x$
3	$50(0.4 \sin(15x^2) + 5x)$
4	$\sin(8\pi x)$
5	$\sin(6\pi x^2)$
6	$\sin(25x + 0.1)/(25x + 0.1)$
7	$2 \sin(2\pi x) + \sin(4\pi x)$
8	$2 \sin(2\pi x) + \sin(4\pi x) + \sin(8\pi x)$
9	$-2 \sin(2\pi x) + \cos(6\pi x)$
10	$2 \sin(2\pi x) + \cos(6\pi x)$
11	$-2 \sin(2\pi x) + \cos(6\pi x) - x$
12	$-2 \cos(2\pi x) - \cos(4\pi x)$
13	$\text{atan}((10x - 5)^3) + 0.5 \text{atan}((10x - 8)^3)$
14	$(4.48x - 1.88) \sin((4.88x - 1.88)^2) + 1$
15	$e^{10x-6} \sin((5x - 2)^2) + (3x - 1)^3$
16	$(1/9) \tanh(9x + 0.5)$

TABLE II: Tested artificial signals.

Fig. 7: Differences for $y = f_1(x)$.Fig. 8: Differences for $y = f_2(x)$.Fig. 9: Differences for $y = f_3(x)$.Fig. 10: Differences for $y = f_4(x)$.

Experiments also proved that application of the CS-RBFs with a constant shape parameter is not convenient, unless additional points are not included in the approximation. (maximum error for the function 2, if CS-RBF is used).

The proposed RBF approximation method was also tested for a newly developed RBF. The experiments proved significant improvement of the final approximation over the TPS function.

Function number	Radial basis function			
	CS-RBF	TPS	Gauss	Proposed
1	$2.41 \cdot 10^{-4}$	$6.40 \cdot 10^{-6}$	$3.13 \cdot 10^{-7}$	$1.21 \cdot 10^{-7}$
2	$1.54 \cdot 10^{-1}$	$3.95 \cdot 10^{-6}$	$2.56 \cdot 10^{-2}$	$2.34 \cdot 10^{-6}$
3	$9.21 \cdot 10^{-4}$	$8.67 \cdot 10^{-6}$	$2.51 \cdot 10^{-4}$	$3.30 \cdot 10^{-7}$
4	$7.92 \cdot 10^{-4}$	$2.56 \cdot 10^{-5}$	$3.12 \cdot 10^{-7}$	$5.23 \cdot 10^{-7}$

TABLE III: Mean square error.

VIII. CONCLUSION

In this contribution a novel approach for RBF approximation based on geometrical properties of a sampled curve (signal) is presented. Experiments proved advantages of the global functions over CS-RBFs are sensitive to the shape parameter

Function number	Radial basis function			
	CS-RBF	TPS	Gauss	Proposed
1	$7.33 \cdot 10^{-2}$	$7.76 \cdot 10^{-3}$	$1.70 \cdot 10^{-3}$	$9.21 \cdot 10^{-4}$
2	$1.81 \cdot 10^0$	$5.24 \cdot 10^{-3}$	$5.19 \cdot 10^{-1}$	$6.29 \cdot 10^{-3}$
3	$1.44 \cdot 10^{-1}$	$1.39 \cdot 10^{-2}$	$5.09 \cdot 10^{-2}$	$2.30 \cdot 10^{-3}$
4	$1.33 \cdot 10^{-1}$	$1.52 \cdot 10^{-2}$	$1.76 \cdot 10^{-3}$	$1.89 \cdot 10^{-3}$

TABLE IV: Maximum absolute error.

Function number	Radial basis function			
	CS-RBF	TPS	Gauss	Proposed
1	$6.60 \cdot 10^{-19}$	$8.80 \cdot 10^{-6}$	$1.44 \cdot 10^{-12}$	$2.49 \cdot 10^{-9}$
2	$2.68 \cdot 10^{-18}$	$8.47 \cdot 10^{-5}$	$2.43 \cdot 10^{-12}$	$1.05 \cdot 10^{-7}$
3	$4.05 \cdot 10^{-18}$	$4.58 \cdot 10^{-6}$	$2.35 \cdot 10^{-12}$	$2.25 \cdot 10^{-9}$
4	$2.78 \cdot 10^{-18}$	$2.44 \cdot 10^{-6}$	$9.17 \cdot 10^{-14}$	$3.20 \cdot 10^{-10}$

TABLE V: Condition numbers.

selection and require more points for acceptable approximation in general.

The newly developed RBF function is better in the precision terms over the TPS function, however, the TPS function has a little bit worse conditionality which can be improved by additional polynomial.

The proposed RBF seems to be an alternative to the TPS function offering better error, however the influence of the shape parameter α is under investigation. The optimal choice of the shape parameter α is an open question.

The experiments also proved that the CS-RBFs require variable shape parameter which is significant result as CS-RBFs are used in many areas, e.g. solution of partial differential equations, etc. The adaptive shape parameter for CS-RBFs is to be explored in future.

IX. ACKNOWLEDGMENT

The authors would like to thank their colleagues and students at the University of West Bohemia for their discussions and suggestions, and especially to Zuzana Majdisova for making some testing functions, Michal Smolik, Marek Zabran and Maria Martynova at the University of West Bohemia for the help with the MATLAB programming issues. Thanks belong also to anonymous reviewers for their valuable comments and hints provided.

REFERENCES

- [1] R. L. Hardy, "Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968-1988," *Computers & Mathematics with Applications*, vol. 19, pp. 163–208, 1990.
- [2] B. Singh and D. Toshniwal, "MOWM: Multiple Overlapping Window Method for RBF based missing value prediction on big data," *Expert Systems with Applications*, vol. 122, pp. 303 – 318, 2019.
- [3] M. Smolik, V. Skala, and O. Nedved, "A comparative study of LOWESS and RBF approximations for visualization," in *International Conference on Computational Science and Its Applications*, 2016, pp. 405–419.
- [4] Z. Majdisova, V. Skala, and M. Smolik, "Algorithm for placement of reference points and choice of an appropriate variable shape parameter for the RBF approximation (accepted for publication)," *Integrated Computer Aided Engineering*, IOS Press, 2019.
- [5] A. Karageorghis and P. Tryfonos, "Shape parameter estimation in RBF function approximation," *International Journal of Computational Methods and Experimental Measurements*, vol. 7, pp. 246–259, 2019.
- [6] J. Wang and G. Liu, "On the optimal shape parameters of radial basis functions used for 2-d meshless methods," *Computer methods in applied mechanics and engineering*, vol. 191, pp. 2611–2630, 2002.
- [7] F. Afiatdoust and M. Esmailbeigi, "Optimal variable shape parameters using genetic algorithm for radial basis function approximation," *Ain Shams Engineering Journal*, vol. 6, pp. 639–647, 2015.
- [8] S. A. Sarra and D. Sturgill, "A random variable shape parameter strategy for radial basis function approximation methods," *Engineering Analysis with Boundary Elements*, vol. 33, pp. 1239–1245, 2009.
- [9] V. Skala, S. Karim, and M. Zabran, "Radial basis function approximation optimal shape parameters estimation: Preliminary experimental results (accepted for publication)," in *SKSM 27 Conference, Selangor, Malaysia, Symposium Kebangsaan Sains ke 27*. AIP Press, 2019.
- [10] M. E. Biancolini, *Fast Radial Basis Functions for Engineering Applications*, 1st ed. Springer International Publishing, 2017.
- [11] G. Fasshauer, *Meshfree Approximation Methods with Matlab*, 1st ed. World Scientific, 2007.
- [12] F. Menandro, "Two new classes of compactly supported radial basis functions for approximation of discrete and continuous data," *Engineering Reports*, vol. 2019;1:e12028, pp. 1–30, 2019.
- [13] M. Smolik and V. Skala, "Large scattered data interpolation with radial basis functions and space subdivision," *Integrated Computer Aided Engineering*, vol. 25, pp. 49–62, 2018.
- [14] —, "Efficient simple large scattered 3D vector fields radial basis function approximation using space subdivision," in *Computational Science and Its Application, ICCSA 2019 proceedings*, 2019, pp. 337–350.
- [15] —, "Classification of critical points using a second order derivative," in *ICCS 2017, Procedia Computer Science*, vol. 108. Elsevier, 2017, pp. 2373–2377.
- [16] —, "Vector field second order derivative approximation and geometrical characteristics," in *ICCSA 2017 Conf.*, vol. 10404. Springer, 2017, pp. 148–158.
- [17] V. Skala, R. Pan, and O. Nedved, "Making 3D replicas using a flatbed scanner and a 3d printer," in *ICCSA 2014*. Springer, 2014, pp. 76–86.
- [18] X. Zhang, K. Z. Song, M. W. Lu, and X. Liu, "Meshless methods based on collocation with radial basis functions," *Computational mechanics*, vol. 26, pp. 333–343, 2000.
- [19] L. Yingwei, N. Sundararajan, and P. Saratchandran, "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm," *IEEE Transactions on neural networks*, vol. 9, pp. 308–318, 1998.
- [20] S. Cuomo, A. Galletti, G. Giunta, and L. Marcellino, "Reconstruction of implicit curves and surfaces via RBF interpolation," *Applied Numerical Mathematics*, vol. 116, pp. 157 – 171, 2017.
- [21] Z. Majdisova and V. Skala, "Radial basis function approximations: comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728–743, 2017.
- [22] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of geophysical research*, vol. 76, pp. 1905–1915, 1971.
- [23] V. Skala, "RBF interpolation and approximation of large span data sets," in *MCSI 2017 Corfu*. IEEE, 2018, pp. 212–218.
- [24] —, "RBF interpolation with CSRBF of large data sets," in *ICCS 2017, Procedia Computer Science*, vol. 108. Elsevier, 2017, pp. 2433–2437.
- [25] J. Jäger, "Advances in radial and spherical basis function interpolation," Ph.D. dissertation, Justus-Liebig-Universitt, Otto-Behaghel-Str. 8, 35394 Gielen, 2018.
- [26] Z. Majdisova and V. Skala, "A radial basis function approximation for large datasets," in *SIGRAD 2016*, 2016, pp. 9–14.
- [27] —, "A new radial basis function approximation with reproduction," in *CGVCVIP 2016*, 2016, pp. 215–222.
- [28] —, "Radial basis function approximations: Comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728–743, 2017.
- [29] —, "Big geo data surface approximation using radial basis functions: A comparative study," *Computers and Geosciences*, vol. 109, pp. 51–58, 2017.
- [30] M. Buhmann, "On quasi-interpolation with radial basis functions," *Journal of Approximation Theory*, vol. 72, no. 1, pp. 103 – 130, 1993.

Reconstruction of Corrupted Vector Fields using Radial Basis Functions

Michal Smolik

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
smolik@kiv.zcu.cz*

Vaclav Skala

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
skala@kiv.zcu.cz*

Abstract—The vector fields may be results from the measurements of real flow experiments. However, during the measurements, some parts of the vector field can be measured incorrectly or even some parts of the vector field are not possible to capture due to some shading and invisibility. In this paper, we focus on the reconstruction of such corrupted vector fields. We detect the locations, where the vector field was measured incorrectly and reconstruct those locations of the vector field. For the reconstruction, we use Radial Basis Functions (RBF) approximation to fill the missing locations of the vector field as well as to correct and smooth the locations of the vector field, where it was probably measured with some error. The results of the proposed method are presented in this paper.

Index Terms—Vector field, reconstruction, Radial basis functions, corrupted data, approximation.

I. INTRODUCTION

The interpolation and approximation are probably the most frequent operations used in computational methods. Several methods have been developed for data interpolation and approximation, but they expect some kind of data "ordering", e.g. structured mesh, rectangular mesh, unstructured mesh, etc. However, in many engineering problems, data are not ordered and they are scattered in d -dimensional space, in general. Usually, in technical applications, the scattered data are tessellated using triangulation but this approach is quite prohibitive for the case of d -dimensional data interpolation because of the computational cost.

Interpolated and approximated scattered vector data on a surface become frequent in applied problem solutions. There are applications for vector field decomposition [1], for vector field design system for surfaces that allows the user to control the number of singularities in the vector field and their placement [2]. The paper [3] uses the vector field interpolation for estimating robust point correspondences between two sets of points. An approach for critical points reduction and vector field approximation using Radial basis functions is presented in [4] with an extension into 3D in [5].

Vector fields can be measured during experiments. However, the accuracy of measurements is sometimes low and

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions. The research was supported by projects Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

sometimes part of the vector field cannot be measured due to obstacle and invisibility of the vector field. There exist some methods that deal with this problem. The paper [6] presents an approach for reconstruction of vector field. Another approach is presented in [7]. The paper [8] presents an approach for reconstruction of wind field from limited information provided by a lidar system. This approach is able to reconstruct complex situations. The follow-up paper [9] presents the possibility of using nacelle-mounted lidar for wind field reconstruction. The reconstruction of 2D vector field from sparse set of points-vectors pairs is presented in [10]. The approach subdivides the domain adaptively in order to make local piece-wise polynomial approximations for the field. After that, it uses the partition of unity to blend the local approximations together. A method for the global vector-field reconstruction of nonlinear dynamical systems from a time series is studied in the paper [11]. It employs a complete set of polynomials and singular value decomposition to estimate a standard function, which is central to the algorithm. The paper [12] presents a method for the reconstruction of flow fields based on adaptive control grid interpolation. The data for reconstruction are acquired from the magnetic resonance and the resulting velocity field is with high quality. The paper [13] aims to show that it is possible to accurately estimate, in a real-time fashion, the radial and tangential velocity components of the wind field from lidar data. The reconstruction is generated through the synthesis of an unscented Kalman filter that employs a low-order dynamic model of the wind to estimate the unmeasured velocities within the wind field, using repeated measurement updates from typical nacelle-mounted lidar instruments. Another approach for reconstruction of vector fields is presented in [14]. This paper focuses on reconstruction of tomography data sets.

In this paper, we present a new approach for reconstruction of incomplete or corrupted vector fields using Radial Basis Functions (RBF).

II. VECTOR FIELD

Vector fields on surfaces are important objects, which appear frequently in scientific simulation in CFD (Computational Fluid Dynamics) or modeling by FEM (Finite Element Method). To be visualized, such vector fields are usually lin-

early approximated for the sake of simplicity and performance considerations.

The vector field can be easily analyzed when having an approximation of the vector field near some location point. The important places to be analyzed are so called critical points. Analyzing the vector field behavior near these points gives us the information about the characteristic of the vector field.

A. Critical Point

Critical points \mathbf{x}_0 of the vector field are points at which the magnitude of the vector vanishes

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

i.e. all components are equal to zero

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2)$$

A critical point is said to be isolated, or simple, if the vector field is non vanishing in an open neighborhood around the critical point. Thus for all surrounding points \mathbf{x}_ϵ of the critical point \mathbf{x}_0 the equation (1) does not apply, i.e.

$$\frac{d\mathbf{x}_\epsilon}{dt} \neq \mathbf{0}, \quad (3)$$

At critical points, the direction of the field line is indeterminate, and they are the only points in the vector field where field lines can intersect (asymptotically). The terms singular point, null point, neutral point or equilibrium point are also frequently used to describe critical points.

These points are important because together with the nearby surrounding vectors, they have more information encoded in them than any such group in the vector field, regarding the total behavior of the field, see Fig. 1.

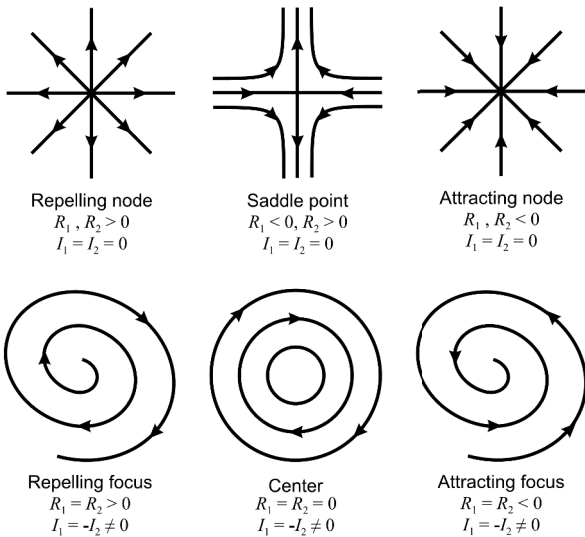


Fig. 1. Classification of 2D first order critical points. R_1, R_2 denote the real parts of the eigenvalues of the Jacobian matrix while I_1, I_2 denote their imaginary parts.

III. RADIAL BASIS FUNCTIONS

The Radial basis functions (RBF) is a technique for scattered data interpolation [15], [16] and approximation [17], [18]. The RBF interpolation and approximation is computationally more expensive, because input data are not ordered and there is no known relation between them. Although the RBF has higher computational cost, it can be used for d -dimensional problem solution in many applications, e.g. solution of partial differential equations, image reconstruction, neural networks, fuzzy systems, GIS systems, optics etc.

The RBF is a function whose value depends only on the distance from some center point. Due to the use of the distance functions, the RBFs can be easily implemented to reconstruct the surface using scattered data in 2D, 3D or higher dimensional spaces. It should be noted that the RBF approximation is not separable.

The RBF interpolation was originally introduced by [19] and is based on computing the distance of two points in the k -dimensional space and is defined by a function

$$f(\mathbf{x}) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (4)$$

where λ_j are weights of the RBFs, M is the number of the radial basis functions, i.e. the number of interpolation points, and φ is the radial basis function. For a given dataset of points with associated values, i.e. in the case of scalar values $\{\mathbf{x}_i, h_i\}_{i=1}^M$, the following linear system of equations is obtained

$$h_i = f(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad \text{for } \forall i \in \{1, \dots, M\} \quad (5)$$

where λ_j are weights to be computed.

Equation (5) can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}, \quad (6)$$

where matrix \mathbf{A} is symmetrical, as $\|\mathbf{x}_i - \mathbf{x}_j\| = \|\mathbf{x}_j - \mathbf{x}_i\|$.

Radial function interpolants have a nice property of being invariant under all Euclidean transformations, i.e. translations, rotations and reflections. It means that it does not matter whether we first compute the RBF interpolation function and then apply a Euclidean transformation, or if we first transform all the data and then compute the radial function interpolants. This is result of the fact that Euclidean transformations are characterized by orthogonal transformation matrices and are therefore 2 norm invariant. Radial basis functions can be

divided into two groups according to their influence. First group are "global" RBF [20], for example:

$$\begin{aligned}
 \text{Thin Plate Spline (TPS)} \quad & \varphi(r) = r^2 \log r \\
 \text{Gauss function} \quad & \varphi(r) = e^{-(\epsilon r)^2} \\
 \text{Inverse Quadric (IQ)} \quad & \varphi(r) = \frac{1}{1 + (\epsilon r)^2} \\
 \text{Inverse Multiquadric (IMQ)} \quad & \varphi(r) = \frac{1}{\sqrt{1 + (\epsilon r)^2}} \\
 \text{Multiquadric (MQ)} \quad & \varphi(r) = \sqrt{1 + (\epsilon r)^2}
 \end{aligned} \quad (7)$$

where ϵ is the shape parameter of radial basis function [21].

The "local" RBF were introduced by [22] as Compactly Supported RBF (CSRBF) and satisfy the following condition

$$\varphi(r) = (1 - r)_+^q P(r) = \begin{cases} (1 - r)^q P(r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (8)$$

where $P(r)$ is a polynomial function and q is a parameter. Typical examples of CSRBF are

$$\begin{aligned}
 \varphi_1(r) &= (1 - \epsilon r)_+ \\
 \varphi_2(r) &= (1 - \epsilon r)_+^3 (3\epsilon r + 1) \\
 \varphi_3(r) &= (1 - \epsilon r)_+^5 (8(\epsilon r)^2 + 5\epsilon r + 1) \\
 \varphi_4(r) &= (1 - \epsilon r)_+^2 \\
 \varphi_5(r) &= (1 - \epsilon r)_+^4 (4\epsilon r + 1) \\
 \varphi_6(r) &= (1 - \epsilon r)_+^6 (35(\epsilon r)^2 + 18\epsilon r + 3) \\
 \varphi_7(r) &= (1 - \epsilon r)_+^8 (32(\epsilon r)^3 + 25(\epsilon r)^2 + 8\epsilon r + 1) \\
 \varphi_8(r) &= (1 - \epsilon r)_+^3 \\
 \varphi_9(r) &= (1 - \epsilon r)_+^3 (5\epsilon r + 1) \\
 \varphi_{10}(r) &= (1 - \epsilon r)_+^7 (16(\epsilon r)^2 + 7\epsilon r + 1)
 \end{aligned} \quad (9)$$

where ϵ is the shape parameter of radial basis function, see Figure 2 for visualization of (9).

The RBF interpolation can be done using "global" or "local" functions. When using "global" radial basis functions the matrix \mathbf{A} will be full, but when using "local" radial basis functions the matrix \mathbf{A} will be sparse, which can be beneficial when solving the system of linear equations $\mathbf{A}\boldsymbol{\lambda} = \mathbf{h}$.

In the case of the vector data, i.e. $\{\mathbf{x}_i, \mathbf{h}_i\}_1^M$ values \mathbf{h}_i are actually vectors, the RBF is to be performed for each coordinate of \mathbf{h}_i .

IV. PROPOSED APPROACH

The purpose of this approach is to reconstruct the measured vector field. During the measurements, there can be some errors, inaccuracies or even some parts of the vector field can be missing. We will consider the flow around a cylinder in the proposed approach¹.

In the first step, we need to locate the missing parts of the vector field. During the measurement, we obtain several vector fields in different timestamps. From all of this vector fields for

¹Data courtesy of Rut Vitkovicova, CTU in Prague, FME, Department of Fluid Dynamics and Thermodynamics, Technická 4, 166 07 Prague, Czech Republic [24].

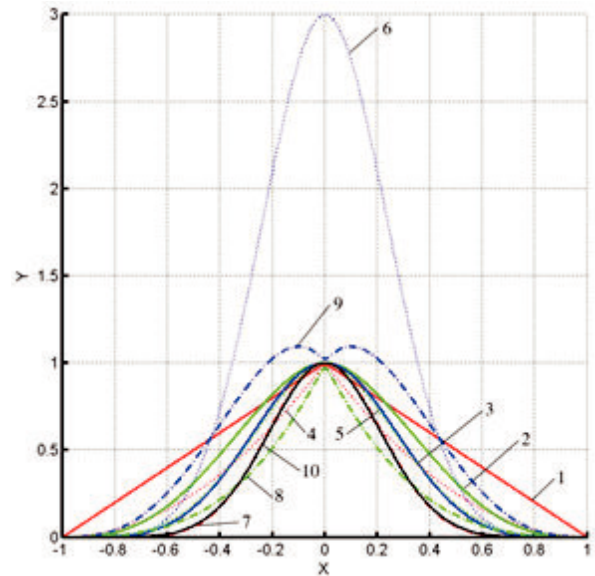


Fig. 2. Examples of CSRBF (from [23]).

different timestamps, we can compute the average speed of the vector field at every location. The parts, where the average speed is low, are measured incorrectly, as the average speed at each location should be similar as we are considering the flow around a cylinder. An example of the computed average speed of the vector field is visualized in Fig. 3. It can be seen, that the low average speed is located at the inflow location on the right side and below the cylinder, where the flow is over-shaded with the cylinder.

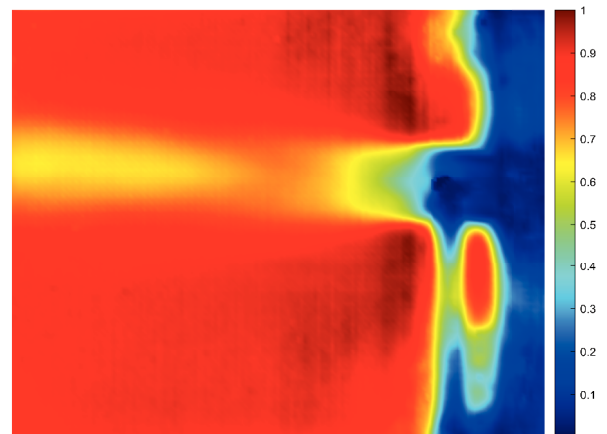


Fig. 3. Average speed of the vector field over simulation time. The color-bar is in relative units [%].

To determine the exact location of incorrectly measured vector field, we need to select some threshold value for the average speed. During the experiments, we determined that the vectors with speed $\leq 30\%$ of the maximal average speed are taken as unmeasured or faulty. The vectors in the interval $(30\%, 50\%)$ may be affected by the error of

measurement. The vectors with speed $\geq 50\%$ are taken as correct, see Fig. 4.

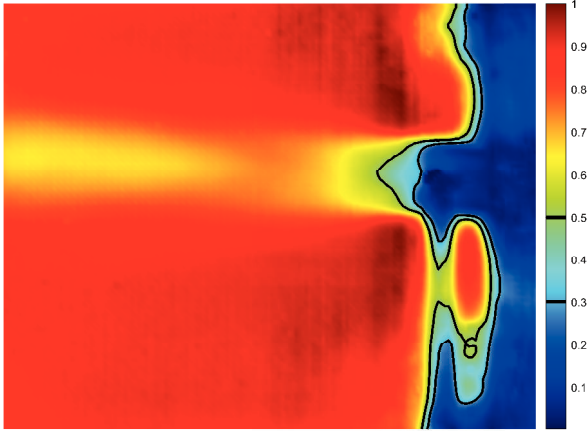


Fig. 4. Average speed of vector field over simulation time with marked contours for 30% and 50% of the maximal average speed. The vectors with speed $\leq 30\%$ are taken as unmeasured or faulty. The vectors with speed in the interval (30%, 50%) may be affected by error. The vectors with speed $\geq 50\%$ are taken as correct.

The correctly measured parts of vector field should be preserved during the reconstruction of corrupted vector field. On the opposite site, the unmeasured or faulty parts of the vector field should be totally removed and replaced with data extrapolation. The vectors from locations, where it is not clearly possible to determine the accuracy of the measured vector field data, can be slightly modified.

To fulfill the requirements described before, we use the approximation method of Radial basis functions. The standard vector field approximation with RBF uses the following equations

$$\mathbf{v}_i = v(\mathbf{x}_i) = \sum_{j=1}^M \lambda_j \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|) \quad \text{for } \forall i \in \{1, \dots, N\}, \quad (10)$$

where $\mathbf{v}_i = [v_i^{(x)}, v_i^{(y)}]^T$ is the vector at \mathbf{x}_i , $\lambda_j = [\lambda_j^{(x)}, \lambda_j^{(y)}]$ is the RBF weight to be computed and $\boldsymbol{\xi}_j$ is the center of the radial basis function.

All the equations (11) forming the system of linear equations have the same importance in the standard RBF approximation. However, we can change the importance or weight of each equation separately. If we multiply one of the equations in (11) as the following

$$\beta_i \mathbf{v}_i = \sum_{j=1}^M \lambda_j \beta_i \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|) \quad \text{for } i \in \{1, \dots, N\}, \quad (11)$$

we end up with the weighted vector field RBF approximation.

If all the coefficients are equal to one, i.e. $\forall i : \beta_i = 1$, then it is as the standard RBF approximation. If we multiply one equation with $\beta_i > 1$, then the i^{th} equation has higher

importance, as during the approximation the i^{th} error is multiplied with number higher than one. The approximation tries to increase the importance at that location and thus approximates the vector field more correctly at \mathbf{x}_i than at other locations (with $\beta_i = 1$). On the opposite side, if we multiply one equation with $0 < \beta_i < 1$, then the i^{th} equation has lower importance, as during the approximation the i^{th} error is multiplied with number smaller than one. Thus, the approximation can differ a bit more at \mathbf{x}_i compared to other locations.

In the proposed approach for a vector field reconstruction, we use the above described weighted vector field RBF approximation. We have defined three types of locations, see Fig. 4. The location with unmeasured or faulty vector field is not used during the vector field approximation. The location with correct vector field is used in the RBF approximation and has the weight $\beta_i = 1$.

The last part of the vector field, that can be affected by error has the weight $\beta_i \in (0, 1)$. To compute the exact value of β_i , we use the following formula

$$\beta_i = \frac{\text{speed}_i - \text{speed}_{START}}{\text{speed}_{END} - \text{speed}_{START}}, \quad (12)$$

where speed_i is the average speed of the vector field at location \mathbf{x}_i , speed_{START} is 30% of the maximal average speed and speed_{END} is 50% of the maximal average speed.

Using this approach, we can compute the RBF approximation of corrupted vector field. However, in the case of Fig. 3, we have no data in the right part of the vector field. This is the inflow location, where the vector field has the direction vectors pointing from left to right. We can use this information and add those vectors at the starting border of the vector field. The speed of the starting vectors is selected as the average speed of all correctly measure vectors.

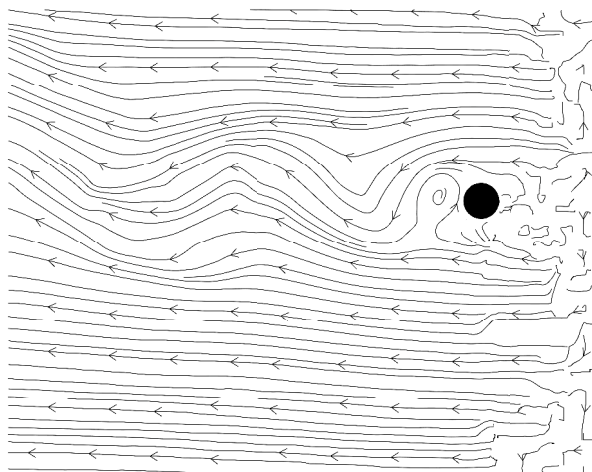
V. EXPERIMENTAL RESULTS

The proposed approach for reconstruction of corrupted vector fields was tested on several measured vector fields. We already presented the result of the average speed of the vector field over the simulation time, see Fig. 3. For the vector field RBF approximation, we used the local radial basis function with the following formula

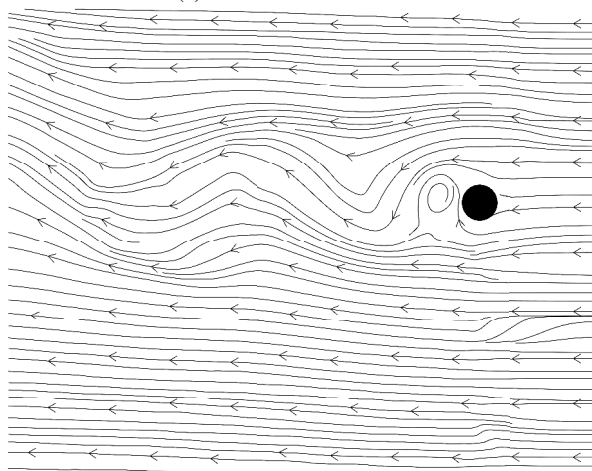
$$\varphi_5(r) = (1 - er)_+^4 (4er + 1). \quad (13)$$

Using the computed average speed (Fig. 3) and our proposed approach, we reconstructed the vector field Fig. 5a. The result after the reconstruction using the proposed approach is visualized in Fig. 5b. It can be seen that the reconstructed vector field has the same important characteristics as the original vector field. The inflow of the vector field has the correct direction beside the direction of the measured inflow vector field.

The computed average speed (Fig. 3), which is used to determine different types of measured vector field (measured correctly, measured incorrectly and measured with possible error) is used for all timestamps of the vector field. Thus when



(a) Measured vector field.



(b) Reconstructed vector field.

Fig. 5. Visualization of measured vector field from the experiment (a) and the corrected vector field with reconstructed missing or error parts (b). Simulation time of this vector field is 5.5 [s].

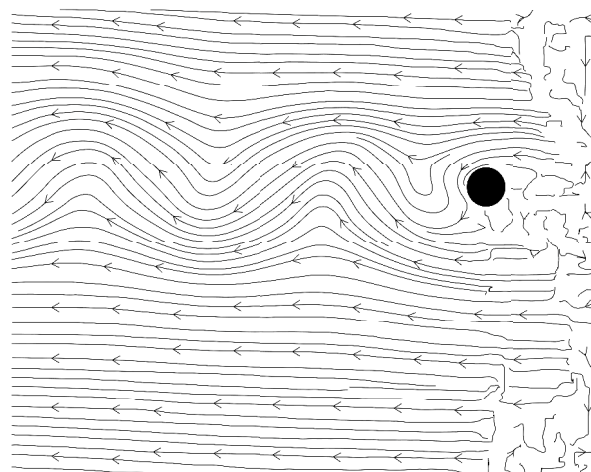
performing the reconstruction of corrupted vector field, the average vector field is computed only once and used for all timestamps of the vector field.

Another result of the proposed approach is visualized in Fig. 6b. This result is the reconstruction of a measured vector field (Fig. 6a) at different time than the previous vector field (Fig. 5a).

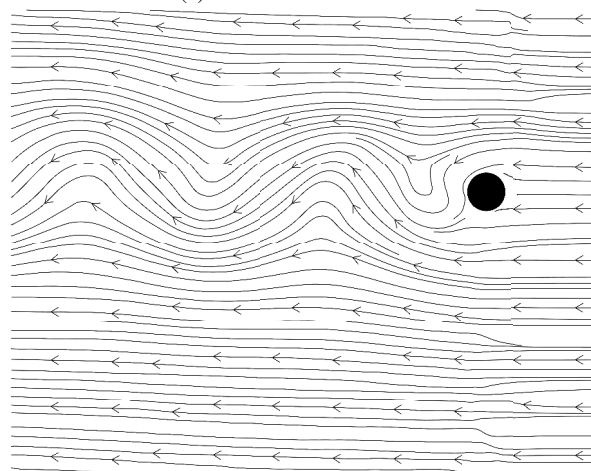
It can be seen, that the vector field reconstruction result is visually correct as the previous example. The proposed approach can be used to reconstruct the missing parts of the measured vector field or to correct incorrectly measured parts of the vector field.

VI. CONCLUSION

We presented a new approach for reconstruction of measured vector fields. The approach reconstructs the missing parts of the vector field and is able to correct incorrectly measured parts of the vector field. As the first step of the pro-



(a) Measured vector field.



(b) Reconstructed vector field.

Fig. 6. Visualization of measured vector field from the experiment (a) and the corrected vector field with reconstructed missing or error parts (b). Simulation time of this vector field is 37 [s].

posed approach, we presented an approach to detect incorrectly or faulty measured parts of the vector field that need to be corrected or reconstructed. For the final reconstruction, we use the introduced weighted Radial basis function approximation method. The presented results proved the ability to reconstruct and correct the measured vector fields.

In the future, we plan to develop even more sophisticated method for detection of incorrectly measured parts of the vector field as well as the extension of the proposed approach to 3D. The extension of the proposed approach to 3D should be straightforward.

ACKNOWLEDGMENT

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions. The research was supported by projects Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

REFERENCES

- [1] G. B. W. Edward J. Fuselier, “Stability and error estimates for vector field interpolation and decomposition on the sphere with rbfs,” *SIAM Journal on Numerical Analysis*, vol. 47, no. 5, pp. 3213–3239, 2009.
- [2] E. Zhang, K. Mischakow, and G. Turk, “Vector field design on surfaces,” *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1294–1326, 2006. [Online]. Available: <http://doi.acm.org/10.1145/1183287.1183290>
- [3] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, “Robust point matching via vector field consensus,” *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1706–1721, 2014. [Online]. Available: <http://dx.doi.org/10.1109/TIP.2014.2307478>
- [4] M. Smolik, V. Skala, and Z. Majdisova, “Vector field radial basis function approximation,” *Advances in Engineering Software*, vol. 123, pp. 117–129, 2018.
- [5] M. Smolik, V. Skala, and M. Zuzana, “3D vector field approximation and critical points reduction using radial basis functions,” in *International Conference on Applied Physics, System Science and Computers*. Springer, 2018, pp. 1–6.
- [6] D. Zotov, A. Shurup, and O. Rumyantseva, “Vector field reconstruction of flows using the novikov–agaltsov functional algorithm and the additive correlation method,” *Bulletin of the Russian Academy of Sciences: Physics*, vol. 81, no. 1, pp. 101–105, 2017.
- [7] L. Bonaventura, A. Iske, and E. Miglio, “Kernel-based vector field reconstruction in computational fluid dynamic models,” *International Journal for Numerical Methods in Fluids*, vol. 66, no. 6, pp. 714–729, 2011.
- [8] D. Schlipf, A. Rettenmeier, F. Haizmann, M. Hofsäß, M. Courtney, and P. W. Cheng, “Model based wind vector field reconstruction from lidar data,” 2012.
- [9] A. Borraccino, D. Schlipf, F. Haizmann, and R. Wagner, “Wind field reconstruction from nacelle-mounted lidars short range measurements,” *Wind Energy Science Discussions*, vol. 2, pp. 269–283, 2017.
- [10] M. Lage, F. Petronetto, A. Paiva, H. Lopes, T. Lewiner, and G. Tavares, “Vector field reconstruction from sparse samples with applications,” in *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2006, pp. 297–306.
- [11] L. Wei-Dong, K. Ren, S. Meunier-Guttin-Cluzel, and G. Gouesbet, “Global vector-field reconstruction of nonlinear dynamical systems from a time series with svd method and validation with lyapunov exponents,” *Chinese Physics*, vol. 12, no. 12, p. 1366, 2003.
- [12] D. Frakes, M. Smith, D. De Zélicourt, K. Pekkan, and A. Yoganathan, “Three-dimensional velocity field reconstruction,” *Journal of biomechanical engineering*, vol. 126, no. 6, pp. 727–735, 2004.
- [13] P. Towers and B. L. Jones, “Real-time wind field reconstruction from lidar measurements using a dynamic wind model and state estimation,” *Wind Energy*, vol. 19, no. 1, pp. 133–150, 2016.
- [14] H. Braun and A. Hauck, “Tomographic reconstruction of vector fields,” *IEEE Transactions on signal processing*, vol. 39, no. 2, pp. 464–471, 1991.
- [15] R. Pan and V. Skala, “A two-level approach to implicit surface modeling with compactly supported radial basis functions,” *Engineering with Computers*, vol. 27, no. 3, pp. 299–307, 2011.
- [16] M. Smolik and V. Skala, “Large scattered data interpolation with radial basis functions and space subdivision,” *Integrated Computer-Aided Engineering*, vol. 25, no. 1, pp. 49–26, 2018.
- [17] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*. World Scientific, 2007, vol. 6.
- [18] V. Skala, “Meshless interpolations for computer graphics, visualization and games,” in *Eurographics 2015 - Tutorials, Zurich, Switzerland, May 4-8, 2015*, 2015. [Online]. Available: <http://dx.doi.org/10.2312/egt.20151046>
- [19] R. L. Hardy, “Multiquadric equations of topography and other irregular surfaces,” *Journal of geophysical research*, vol. 76, no. 8, pp. 1905–1915, 1971.
- [20] I. Schagen, “Interpolation in two dimensions—a new technique,” *IMA Journal of Applied Mathematics*, vol. 23, no. 1, pp. 53–59, 1979.
- [21] B. Fornberg and C. Piret, “On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere,” *J. Comput. Physics*, vol. 227, no. 5, pp. 2758–2780, 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.jcp.2007.11.016>
- [22] H. Wendland, “Computational aspects of radial basis function approximation,” *Studies in Computational Mathematics*, vol. 12, pp. 231–256, 2006.
- [23] K. Uhler and V. Skala, “Radial basis function use for the restoration of damaged images,” in *Computer vision and graphics*. Springer, 2006, pp. 839–844.
- [24] Y. Yokoi and R. Vitkovičová, “Experimental investigation of the mutual interference flow of two circular cylinders by flow visualization,” in *EPJ Web of Conferences*, vol. 143. EDP Sciences, 2017, p. 02146.

Towards Complete SAT-based Cryptanalysis of RC5 Cipher

1st Artur Sobon
HSBC Service Delivery
Developers Division
 Kraków, Poland
 artur.sobon@gmail.com

2nd Mirosław Kurkowski
Cardinal St. Wyszyński University
Institute of Computer Science
 Warszawa, Poland
 m.kurkowski@uksw.edu.pl

3rd Sylwia Stachowiak
Rozprza Primary School
Division of Mathematics
 Piotrków Trybunalski, Poland
 sylwiastachowiak@interia.eu

Abstract—There are many concepts and methods of ciphers cryptanalysis. It depends on the type of the cipher investigated or sometimes modes of operations. Usually, cryptanalysts would like to check some important properties of well known and used ciphers, where so far there is no possibility to break them. Investigators try sometimes break a weaker version of the cipher for understanding which and how proper cipher's parts are cryptographically strong. SAT-based cryptanalysis is one of the used and efficient ways to investigate important properties of some symmetric ciphers. In this paper, we show our investigations and new experimental results in the case of SAT-based, direct cryptanalysis of the RC5 cipher. For this solution, having a given cipher, we firstly built a propositional logical formula that encodes the RC5 algorithm. Next, we randomly generate plaintext and a key. Then we compute the ciphertext. At the end of our computations, we use SAT-solvers, specially designed tools for checking about satisfiability of the Boolean formulas. For our experiments, we explore RC5 properties in the case of cryptanalysis with plaintext and ciphertext. In our investigations, we use and compare several SAT solvers: a few new ones and a few rather old but still efficient. We present our results in the case of a little bit weaker version of the RC5 cipher.

Keywords: Symmetric ciphers, satisfiability, SAT based cryptanalysis.

I. INTRODUCTION

SAT - boolean satisfiability problem is probably the most famous NP-complete problem. SAT can be successfully adapted to solving many problems that can be encoded as the propositional Boolean formula [2]. Of course, often these formulas are very huge and in the whole case, solving satisfiability is very hard. However, the satisfiability of many formulas with hundreds or thousands of variables can be done sometimes surprisingly quickly. Almost all of the implemented algorithms for SAT computations are modified and/or optimized versions of the well known DPLL scheme [9], [10]. For solving satisfiability special programs/tools, called SAT-solvers, have been introduced. Of course, none of them can solve all big formulas consist of thousands of variables because in the worst case there is a need to search a big binary tree with depth over a few thousand. But these tools can often answer the question about boolean satisfiability, even for big formulas. SAT-solvers takes as input the conjunctive normal form of the formula investigated (CNF). It is a conjunction of clauses,

where a clause is a disjunction of literals, and a literal is a propositional variable or its negation.

SAT procedures among others are successfully used for cryptanalysis of some cryptographic algorithms or their parts/modifications, especially symmetric ciphers. This idea supplements other classic methods of cryptanalysis [5], [7], [20], [21]. In the literature can be still found new papers devoted to this topic [8], [12], [13], [16], [17], [18], [19], [22], [23].

In this work, we continue ideas introduced in [8], [12], where the authors showed the efficiency of SAT-based cryptanalysis of the Feistel Network and some modifications of the DES cipher. Here we try to increase investigations in this area considering next, well-known and widely used Rivest's cipher: RC5. We try to check how SAT solvers work with the whole and some modifications of the RC5 cipher. We also checked and compared how several new SAT solvers work in this case.

The rest of this paper is organized as follows. In Section 2, we present all needed, basic information on Feistel and RC5 ciphers. We do this, to the extent necessary for explaining our ideas and boolean encoding. Section 3 gives an algorithm of a direct, boolean encoding of the ciphers investigated. In Section 4, we present experimental results as we have obtained. At the end of the paper, we will give some conclusion and future directions connected to our research.

II. FEISTEL NETWORK AND RC5 CIPHER

Now we will present essential knowledge about symmetric ciphers: Feistel Network and RC5 cipher to understand our methodology of SAT-based cryptanalysis used in our research.

A. Feistel Network

The Feistel Network (FN) is one of block symmetric ciphers that was introduced in 1974 by Horst Feistel in the IBM Lab. Initially, FN was used as a part of IBM's cipher named Lucifer, designed by Feistel and Coppersmith. Thanks to special, iterative character of FN, implementing the cipher in hardware tools is very easy. It is worth to note that the simple structure of Feistel Network provides using Feistel-like networks to design numerous widely used and important ciphers. Good examples are: DES, MISTY1, Skijack, early

mentioned Lucifer, Blowfish [21] and RC5. Below we present the basic concept of this algorithm.

Let F be the so-called round function and K_1, \dots, K_n denote a sequence of keys obtained in some way from the main key K for the rounds $1, \dots, n$, respectively. We use the symbol \oplus for denoting the XOR (eXclusive-OR) operation.

The basic operations of FN are the following:

1. break the plaintext block into two equal length parts denoted by (L_0, R_0) ,
2. for each round $i = 0, \dots, n$, compute $L_{i+1} = R_i$ and $R_{i+1} = L_i \oplus F(R_i, K_i)$.

Then the ciphertext is (R_{n+1}, L_{n+1}) .

The structure of FN allows an easy method of decryption. Lets recall basic properties of operation \oplus for all $x, y, z \in \{0, 1\}$:

- $x \oplus x = 0$,
- $x \oplus 0 = x$,
- $x \oplus (y \oplus z) = (x \oplus y) \oplus z$.

A proper ciphertext (R_{n+1}, L_{n+1}) can be decrypted by the following way $R_i = L_{i+1}$ and $L_i = R_{i+1} \oplus F(L_{i+1}, K_i)$, for $i = n, \dots, 0$. It is easy to observe that (L_0, R_0) is the plaintext again. Observe that we have more such equations:

$$\begin{aligned} R_{i+1} \oplus F(L_{i+1}, K_i) &= (L_i \oplus F(R_i, K_i)) \oplus F(L_i, K_i) = \\ &= L_i \oplus (F(R_i, K_i) \oplus F(L_i, K_i)) = L_i \oplus 0 = L_i. \end{aligned}$$

Now we will discuss one of the algorithms developed by Ronald Rivest. These algorithms are briefly denoted by RC (Ron's Ciphers or Rivest's Ciphers).

B. RC5 algorithm

RC5 was published in 1994. When designing the algorithm, Rivest set himself the following goals [15].

- The algorithm should be easy in implementation with regard to software and hardware.
- The algorithm should be rather fast (therefore RC5 uses only operations on bits-blocks which correspond to words in terms of hardware, e.g., when to run on 32-bit processors, the algorithm executes operations on 32-bit blocks).
- The algorithm should be flexible as concerns encryption parameters used (sizes of a block and a key, as well as the number of rounds, should be modifiable). The user should be able to set cipher parameters in order to have control over the relation between the algorithm's efficiency and its security level.
- The algorithm should be as simple as possible so that the cryptographic strength of RC5 can be rapidly determined.
- The algorithm should have low memory requirements, so that it may easily be implemented, e.g., on smart cards.
- The algorithm should provide a suitably high security level.

It is important to note that Rivest has designed a cipher that could be suitably adapted to the user's needs. It allows hard enciphering - big parameters (big key length, block length, number of rounds) for governments, military solutions, etc. or rather weak enciphering (for the aunt that encrypting the cake recipe).

The algorithm uses three kinds of operations: addition modulo n , addition modulo 2^{32} , and permutations (cyclic shifts, to be precise). It is also possible applying plaintext blocks of variable length, but in the example presented below we use of 64-bit data blocks. The number of algorithm cycles can also be different; we denote it by r . As in the previous case, also here the algorithm uses a basic key, on which 32-bit auxiliary keys used for plaintext processing are based. There are $2r + 2$ such keys; we denote them by $S(0), S(1), S(2), \dots, S(2r + 1)$.

In the encryption process, the plaintext block is first divided into 32-bit words: A and B . Then, by means of keys $S(i)$ ($i = 1, \dots, 2r + 1$), the algorithm perform the following operations:

```
A = A + S(0)
B = B + S(1)
For i = 1 to r
  A = ((A XOR B) <<< B) + S(2i)
  B = ((B XOR A) <<< A) + S(2i+1)
```

The cryptogram is a concatenation of blocks A and B .

In the notation presented above, the symbol $>>>$ means a cyclic bit-shift to the right (analogously, $<<<$ means a cyclic bit-shift to the left). All addition operations in the cipher are performed modulo 2^{32} .

The decryption method is equally easy. The ciphertext block is divided into two parts, A and B , and then the following operations should be executed:

```
For i = r downto 1
  B = ((B - S(2i+1)) >>> A) XOR A
  A = ((A - S(2i)) >>> B) XOR B
  B = B - S(1)
  A = A - S(0)
```

We also mean here, that the symbol $>>>$ denotes a bit-shift and all subtraction operations should be performed modulo 2^{32} .

The creating process of the vector $S(i)$ of keys is a little more complicated. Firstly, bytes of the key are copied to a vector L with c 32-bit words (if needed, any unfilled byte positions of the last word are zeroed). Then, the vector S , which applies a linear generator modulo 2^{32} , is initialized.

```
S(0) = P
For i = 1 to 2(r+1)
  S(i) = (S(i-1) + Q)
```

In the RC5 cipher there are two different constants $P = b7e15163$ and $Q = 9e3779b9$ that are counterparts of the values e and ϕ numbers written in the hexadecimal representation. Once again, addition action is performed modulo 2^{32} .

Finally, we apply S to L (remember that n is the greater of two values $2(r + 1)$ and c):

```
i = j = 0
A = B = 0
For i = 1 to 3n
  A = S(i) = (S(i) + A + B) <<< 3
  B = L(j) = (L(j) + A + B) <<< (A+B)
```

$$\begin{aligned} i &= (i + 1) \bmod 2(r+1) \\ j &= (j + 1) \bmod c \end{aligned}$$

Summarizing it should be mention, RC5 can be treated as a family of algorithms. In the example presented, we used 32-bit words and a 64-bit block. It is possible, however, to use 64-bit words and a 128-bit block and others. For $w = 64$ values of constants P and Q are equal to $7e151628aed2a6b$ and $9e3779b97f4c15$ respectively. Rivest has marked different algorithm modes by $RC5-w/r/b$, where w is the size of a word, r is the number of cycles, and b is the size of the key given in bytes.

RSA Laboratories devoted a lot of time for the practical cryptanalysis of the full version that uses 64-bit blocks. After five rounds the computed statistics seem to be very good. After eight rounds each bit of plaintext is the subject of at least one shift. An attack by means of differential cryptanalysis in the case of five cycles of the RC5 algorithm requires 2^{34} plaintexts, 10 cycles require 2^{45} , 12 cycles require 2^{53} , and 15 cycles require 2^{68} plaintexts. Investigations with the use of linear cryptanalysis showed that the algorithm provides an appropriate level of security for more than six cycles. Ron Rivest recommended the use of 12 or even 16 cycles, however, this number may grow with increases in computational power.

C. RC5-breaking project

The following story is very important and instructive. In 1997 RSA Security Inc. has been proposed a challenge to break several symmetric ciphers, including RC5. The company has offered a \$10.000 prize for breaking any of them. In the same year, an organization called *distributed.net* [11], aimed at the brute-force breaking of the RC5-32/12/7 cipher (with a 56, 64 and 72-bit key), was founded. It was one of the first distributed processing projects ever. The concept of *distributed.net* was to use thousands of PC-s machines for creating a computer network with huge computational power. Users (from all over the world) of this network should install a small application, which analyzes downloaded data blocks whenever a computer is idle. By now, cryptograms encrypted with 56- and 64-bit keys have been already cracked. Currently, the project aims at breaking RC5 with a 72-bit key [15], [11].

Here we present some results obtained by *distributed.net*. The network broke the RC5 cipher with a 56-bit key length after 250 days of computation. During this competition, 26 325 130 699 341 825 keys have been tested (36.5% of the total value of the keyspace). Then the network started breaking of the 64-bit key cipher. It took 1757 days with a computing of 15 268 315 356 922 380 000 keys (82.7% of the total value of the keyspace). Today (7th of September 2019) there is 6 122nd day of the RC5-72 breaking competition. So far 286 699 398 233 042 918 393 keys have been tested, which is 6.05 % of the keyspace.

All of the key operations in RC5, from boolean encoding point of view, can be defined by some equivalences (i.e. permutations, rotations, expansions). In the next section, we will describe the full encoding process of this cipher.

III. BOOLEAN ENCODING FOR CRYPTANALYSIS

Using the characterizations presented above now we can show the method of direct, boolean encoding of the two ciphers mentioned. Here we follow for proposed in [12] methodology, where each of bits during cipher executing is encoded as propositional Boolean variable. Here we first show how we can encode the FN cipher. Then, we present the encoding of the main steps of RC5.

As before in this work, we deal with the Feistel Network with a 64-bit block of plaintext and a 32-bit key. Let the propositional variables representing a plaintext bits, a key, and the ciphertext be described as p_1, \dots, p_{64} , k_1, \dots, k_{32} and c_1, \dots, c_{64} respectively. Observe that follow the Feistel algorithm the first half of ciphertext can be encoded as¹:

$$\bigwedge_{i=1}^{32} (c_i \Leftrightarrow p_{i+32}). \quad (1)$$

As a simple example of function F (occurred in FN), we use XOR operator (denoted by \oplus). (It is obvious that this is one of the simplest possible instantiations of function F , but at this point, we try to only show our encoding idea for the whole FN structure.) As it can be seen that for the second half of ciphertext we have the following formula:

$$\bigwedge_{i=33}^{64} (c_i \Leftrightarrow (p_{i-32} \oplus k_{i-32} \oplus p_i)). \quad (2)$$

Now, we can encode the formula for the whole round of FN as follows:

$$\Phi_{FN}^1 : \bigwedge_{i=1}^{32} (c_i \Leftrightarrow p_{i+32}) \wedge \bigwedge_{i=33}^{64} (c_i \Leftrightarrow (p_{i-32} \oplus k_{i-32} \oplus p_i)). \quad (3)$$

Here we can investigate the case of j rounds of FN cipher. Let (p_1^1, \dots, p_{64}^1) , (k_1, \dots, k_{32}) are a plaintext and a key vectors of variables, respectively. By (p_1^k, \dots, p_{64}^k) and (c_1^i, \dots, c_{64}^i) we describe vectors of variables representing input of k -th round for $k = 2, \dots, j$ and output of i -th round for $i = 1, \dots, t-1$. We denote by (c_1^j, \dots, c_{64}^j) the variables of a cipher vector after j -th round, respectively.

The formula which encodes the whole j -th round of a Feistel Network is as follows:

$$\begin{aligned} \Phi_{FN}^j : & \bigwedge_{i=1}^{32} \bigwedge_{s=1}^j (c_i^s \Leftrightarrow p_{i+32}^s) \wedge \bigwedge_{i=33}^{64} \bigwedge_{s=1}^j [c_i^s \Leftrightarrow (p_{i-32}^s \oplus k_{i-32} \oplus p_i^s)] \wedge \\ & \wedge \bigwedge_{i=1}^{64} \bigwedge_{s=1}^{j-1} (p_i^{s+1} \Leftrightarrow c_i^s). \end{aligned} \quad (4)$$

It is easy to observe that the final part of the formula guarantees that the output bits from s -th rounds are the inputs bits in the $(s+1)$ -th.

¹In the next considerations we use propositional logic language with the following connectives: \vee for denoting the alternative and \wedge for denoting conjunction, \Rightarrow implication, \Leftrightarrow for equivalence, \bigwedge , and \bigvee denote finite conjunctions and alternatives respectively.

Observe, the formula obtained using this encoding is a conjunction of simple equivalences. It is very important due to translating into CNF. The next advantage of this encoding scheme is that we can fully automatically generate the formula for many cipher rounds investigated.

As it can be seen, in the case of RC5 cipher, almost all operations done during cipher executions are substitutions, adding modulo 2^{32} , XOR or rotations. All these operations can be fully encoded into a propositional Boolean formula. Here we show an encoding procedure in some detail of the most important parts only for the cipher. An advantage of our idea is a strict, direct encoding of each bit that occurs in the process of the RC5 execution, with no redundancy from the size of the encoding formula point of view. For describing each different bit in this procedure we use one different propositional variable. We can encode directly all parts of the RC5 cipher.

The whole structure of the encoding formula, in the case of RC5 encoding, is similar to FN-example presented above. We can consider RC5 as a sequence of permutations, XORs, additions and key bits rotations. Each of these operations can be encoded as a conjunction of propositional equivalences or implications in the following way.

For example, if we mark by ρ the permutation over the set $\{1, 2, \dots, 64\}$ (i.e. $\rho : \{1, 2, \dots, 64\} \rightarrow \{1, 2, \dots, 64\}$) as the permutation function of RC5 and by $(q_1, \dots, q_{64}), (p_1, \dots, p_{64})$ be a sequence of output and input variables respectively it is easy to observe, that we can encode ρ using the following formula:

$$\bigwedge_{i=1}^{64} (q_i \Leftrightarrow p_{\rho(i)}). \quad (5)$$

Similarly, we can encode all the permutations, and rotations used by RC5.

Another problem occurs in the case of binary addition modulo 2^{32} . Here we explain our encoding method for this mathematical operation. Binary addition follows the same rules as the addition in the decimal system except that rather than carrying a 1 over when the values added equal 10, carry over occurs when the result of addition equals 2. Refer to the example below for clarification. Note that in the binary system:

$$\begin{aligned} 0 + 0 &= 0, \\ 0 + 1 &= 1, \\ 1 + 0 &= 1, \\ 1 + 1 &= 0, \text{ carry over the } 1, \text{ i.e. } (10)_{bin}. \end{aligned}$$

Let's assume that a variable p denotes the result of an addition $a + b$, the letter c is used for denoting a carry. The next variables from q to u are used as auxiliary variables. Also, we use the symbol \oplus for denoting the exclusive-OR operation (XOR).

We can denote an addition operation in the logical language as:

$$\begin{aligned} p &\Leftrightarrow (a \oplus b) \oplus c, \\ \text{where } c &\Leftrightarrow (a \wedge b) \vee (a \wedge c) \vee (b \wedge c). \end{aligned}$$

Note that we can convert these formulas to conjunction with the auxiliary factors in the following way:

$$\begin{aligned} q &\Leftrightarrow (a \oplus b), \\ p &\Leftrightarrow (q \oplus c), \\ r &\Leftrightarrow (a \wedge b), \\ s &\Leftrightarrow (a \wedge c), \\ t &\Leftrightarrow (b \wedge c), \\ u &\Leftrightarrow (r \vee s), \\ c &\Leftrightarrow (u \vee t). \end{aligned}$$

With these notations, we can see, that we are able to easily convert it to CNF form and use it in SAT Solver.

As we have shown before using such procedures, we can encode any given number of rounds of RC5 cipher into a propositional Boolean formula. It is important to note, that from the computational point of view, it is significant to decrease as far as possible the number of variables and connectives used in the formula. So having many equivalences in the formula encoded we can remove redundant ones from the formula.

The cryptanalysis procedure we propose in this paper is the following. Firstly we encode a single round of the cipher considered as a boolean propositional formula. Then the formula encoding the desired number of iteration rounds (or the whole cipher) is automatically generated. Next, we convert the formula obtained into CNF. Here we randomly choose a plaintext and the key vector as a 0, 1-valuation of the variables representing them in the formula. Next, the chosen valuation into the formula is inserted. Now we calculate the corresponding ciphertext using an appropriate key and insert it into the formula. Finally, we run SAT-solver with the plaintext and its ciphertext bits inserted, to find a satisfying valuation of the key variables.

IV. EXPERIMENTAL RESULTS

For our experiments, we chose several from the big set of SAT-solvers. All of them were at the top of SAT-solver Competitions.

A. SAT-solvers

The International SAT Solver Competition *SATRace* is today an early organized series of competitive events aiming at objectively evaluating the progress in state-of-the-art procedures for solving Boolean satisfiability (SAT) instances. Over the years, the competitions have significantly contributed to the fast progress in SAT solver technology that has made SAT a practical success story of computer science. [6].

In our experiments, we have decided to use a few of the most popular and recognized SAT solvers (like MiniSAT). These SAT-solvers were also used by us in previous works [8], [12]. Solvers we choose were the best ones taking part in SAT Competitions.

The best results for the largest number of rounds were obtained by CaDiCal, RSat, and PicoSAT. In all cases, the worst turned out to be SAT4j. It is worth noting that the MiniSAT that achieved high positions in the SAT Competition

12 years ago (SAT 2007 Competition), for this problem obtained worse results, but not the worst ones.

Here we briefly present some most important information about the chosen SAT solvers. It is important to note that they were often awarded the past several years in the mentioned competition *SATRace*.

glu_vc is a SAT solver submitted to the hacking track of the SAT Competition 2017. It updates Glucose 3.0 in the following aspects: phase selection, learned clause database reduction and decision variable selection [6].

Lingeling is a SAT solver proposed by scientists on Johannes Kepler University (JKU) in Linz [3]. It uses some techniques to save space by reduction of some literals [4]. The first time it was presented on SAT Competition in 2010. Through years it has been developing and the latest version was presented on SAT competition in 2013.

PicoSat was also created on JKU [1]. It has many similar solutions as *MiniSAT 1.14*, which is a well-known SAT solver. First-time was shown in 2007. Low-level optimization saves memory and efficiently increase this solver power.

CaDiCal, created on JKU too. It's a solver originally developed to simplify the design of internal data structures [14]. The first time it was presented in 2017 on *SATRace* and it's the latest created SAT solver from JKU considered in this paper.

RSat, uses a phase selection heuristic that is oriented toward reducing work repetition and a frequent restart policy. *RSat* won gold medals from the SAT'07 competition in the industrial category. The previous version of *RSat* also won third place in *SAT-Race 2006*.

MiniSat, started out 2003 as an effort to help people get into the SAT community by providing a small, yet efficient, SAT solver with good documentation. In later versions, the codebase has grown a bit to encompass recent improvements but is still quite small and hopefully readable. In the SAT competition 2005, version 1.13 proved that *MiniSat* still is state-of-the-art; at least for publically available solvers.

Sat4j, is a Java library for solving boolean satisfaction and optimization problems. It can solve SAT, MAXSAT, Pseudo-Boolean, Minimally Unsatisfiable Subset (MUS) problems. Being in Java, the promise is not to be the fastest one to solve those problems (a SAT solver in Java is about 3.25 times slower than its counterpart in C++).

Detailed information about SAT-solvers versions used are the following: *MiniSAT* ver. 2.2.1-5build1, *SAT4j* ver. 2.3.5, *PicoSAT* ver. 960, *Glucose Syrup* ver. 4.1, *RSat* 2.01, *Lingeling* ver. bcj78x, *CaDiCal* ver. 1.0.3.

B. Cryptanalysis procedure

In this research we use the cryptanalysis algorithm proposed in [12] and usefully used in [8]. We follow this procedure:

- 1) we encode the one i -th round of the cipher as a boolean propositional formula,
- 2) we automatically generate the cipher complete formula for many iterations of rounds (the whole cipher),
- 3) we convert the formula obtained into a CNF form,

- 4) we randomly choose plaintext bits and a key vector as a valuation of proper propositional variables in the formula,
- 5) we add mentioned above valuation into the formula,
- 6) we use SAT-solver for calculating a satisfaction valuation including a valuation of ciphertext variables,
- 7) we remove the key valuation from the formula and add the ciphertext one,
- 8) using SAT-solver we calculate a satisfaction valuation including a valuation of the key variables.

As we mentioned before such a procedure was successfully used in [12] and [8]. Then the DES algorithm and its modifications were investigated. Here this method could break almost 6 rounds of the DES algorithm (with 28 key bits added into the formula). It was also shown that DES without S-boxes can be broken easily with 16 rounds in a few seconds. This is confirmation of how important are S-boxes in the ciphers. The most important property of S-boxes power seems to be their nonlinearity. Experiments done in [8] showed that DES with linear S-boxes can be broken as fast as DES without S-boxes.

So that is the reason that it is important to investigate the properties of weaker versions of the ciphers.

C. Simplification

As we mentioned before all parts of the RC5 cipher can be encoded as a Boolean propositional formula. A sketch of such encoding was done in previous parts of the paper. Unfortunately, there is a problem with rotations. RC5 uses a value-depending rotation. This type of rotation depends on the value of the variable used at the proper point in the execution of RC5. We have an idea of how we can encode such rotation, but in this paper, we simplify our consideration and use a little bit simplified version of the cryptanalysis procedure.

As we have written above generally, SAT-based cryptanalysis depends on a formula that encodes all behaviors of events taking place during cipher execution. In our computations we use the following simplification:

After choosing bits of plaintext and key we compute the ciphertext parallelly on an ordinary software. This implementation help us to know about the values of RC5 rotations during the execution of the cipher with chosen values of the plaintext and the key.

We add this information into the formula we generate for several rounds of RC5. The results obtained are the following.

In the tables presented below, we can see our experimental results.

First, we have computed formulas for 1,4,8,12,16, and 20 rounds of RC5 cipher according to the assumptions presented above. Obtained results seem to be rather natural and similar to dimensions of formulas obtained in the case of the DES cipher.

For 20 rounds of RC5, we have obtained over 13 hundreds of variables and less than 40 thousand clauses. The numbers of variables are not so big for SAT-solvers, and we were hoped that computations will finish successfully.

Round	Size of S [b]	Variables	Clauses
1	128	1216	3448
2	192	1856	5300
4	320	3136	9004
8	576	5696	16412
12	832	8256	23820
16	1088	10816	31228
20	1344	13376	38636

TABLE I
FORMULAS SIZES

Next, we compute our formulas using different SAT-solvers. Taking into account our simplification getting good results was not surprising. Almost all the results were less than 0.5 sec. Most of them less than 0.01 sec.

SAT Solver	Time [ms]	Time [ms]	Time [ms]
	1 round	4 round	8 round
RSat	1	2	7
PicoSAT	3	6	12
CaDiCal	3	11	13
Glucose Syrup	4	8	19
MiniSAT	9	13	23
LingeLing	10	34	46
SAT4j	327	371	358

SAT Solver	Time [ms]	Time [ms]	Time [ms]
	12 round	16 round	20 round
CaDiCal	17	18	21
RSat	9	42	24
PicoSAT	18	22	29
MiniSAT	30	42	56
Glucose	20	46	75
LingeLing	55	89	99
SAT4j	405	486	509

TABLE II
RESULTS FOR DIFFERENT SOLVERS AND ROUNDS

Our experiments confirmed that value-dependent rotations are the key point of the RC5 cipher. Without these parts of the algorithm, the cipher can be broken easily.

All our experiments were carried out in the environment Linux, version 5.0.0-29-generic. The physical machine was equipped with 1 core (1 logical CPU) processor from the Intel Haswell Family - Intel(R) Core(TM) i7-7660U CPU @ 2.50GHz with 4MB SmartCache.

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have presented our research in the case of SAT-based, direct cryptanalysis of the RC5 cipher. We showed and compare results obtained from several efficient SAT-solvers. We have checked how the solvers work in the case of cryptanalysis and how they solve some problems with some simplification of the RC5 cipher.

Interesting seems to be our observation that the RC5 without value-dependent rotations is a very weak cipher and can be broken in the time less than 1 sec.

In our next work, we will try to construct a formula that encodes the whole cipher with value-dependent rotations and check how SAT-solvers can break this formula. Next, we will try to apply our experience for the SAT cryptanalysis of several other ciphers like Blowfish, Twofish, AES, or hash functions.

VI. ACKNOWLEDGEMENTS

The first and the second author would like to thanks Renata Czerwik (Jędrasiak) for her help in motivation for our current cooperation and in this research.

REFERENCES

- [1] A. Biere, *PicoSAT Essentials*. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, vol. 4, pp. 75 – 97, Delft University, 2008.
- [2] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [3] A. Biere, *Lingeling, Plingeling, Picosat and Precosat at SAT Race 2010*. Technical Report FMV Reports Series 10/1, Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria, 2010.
- [4] A. Biere, *Lingeling, Plingeling and Treengeling Entering the SAT Competition 2013*. In *Proceedings of SAT Competition 2013*, A. Balint, A. Belov, M. Heule, M. Jarvisalo (editors), vol. B-2013-1 of Department of Computer Science Series of Publications B, pages 51-52, University of Helsinki, 2013.
- [5] E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*. *J. Cryptology*, 4(1):3-72, 1991.
- [6] J. Chen, *Proceedings of SAT Competition 2017: Solver and Benchmark Descriptions*, volume B-2017-1 of Department of Computer Science Series of Publications B, University of Helsinki, 2017.
- [7] N. Courtois and G. V. Bard, *Algebraic cryptanalysis of the Data Encryption Standard*. In S.D. Galbraith, editor, *IMA Int. Conf.*, volume 4887 of *Lecture Notes in Computer Science*, pages 152-169. Springer, 2007.
- [8] M. Chowaniec, M. Kurkowski, and M. Mazur, *New Results in Direct SAT-Based Cryptanalysis of DES-Like Ciphers*. In: Pejaś J., El Fray I., Hyla T., Kacprzyk J. (eds) *Advances in Soft and Hard Computing*. ACS 2018. *Advances in Intelligent Systems and Computing*, pp. 282-294, vol 889. Springer, Cham.
- [9] M. Davis and H. Putnam, *A computing procedure for quantification theory*. *J. ACM*, 7(3):201-215, 1960.
- [10] M. Davis, G. Logemann, and D. W. Loveland, *A machine program for theorem-proving*. *Commun. ACM*, 5(7):394-397, 1962.
- [11] <https://distributed.net> (accessed on 7th of September 2019).
- [12] P. Dudek, M. Kurkowski, and M. Srebrny, *Towards Parallel Direct SAT-based Cryptanalysis*, in *PPAM'11 Proceedings*, pp. 266-275, vol. 7203 of LNCS, Springer Verlag, 2012.
- [13] A. D. Dwivedi, et al., *SAT-based Cryptanalysis of Authenticated Ciphers from the CAESAR Competition*, in *Proc. of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017)* - vol. 4: SECRIPT, pp. 237 – 246, 2017.
- [14] <https://github.com/arminbiere/cadical> (accessed on 7th of September 2019).
- [15] Cz. Kościelny, M. Kurkowski, M. Srebrny, *Modern Cryptography Primer*. Springer Verlag, 2013.
- [16] F. Lafitte, L. Lerman, O. Markowitch, and D. van Heule, *SAT-based cryptanalysis of ACORN*, *IACR Cryptology ePrint Archive*, pp. 521, vol. 2016, 2016.
- [17] F. Lafitte, J. Nakahara Jr., and D. van Heule, *Applications of SAT Solvers in Cryptanalysis: Finding Weak Keys and Preimages*, *JSAT*, vol. 9, pp. 1-25, 2014.
- [18] F. Massacci, *Using Walk-SAT and Rel-SAT for cryptographic key search*. In T. Dean, editor, *IJCAI*, pages 290-295. Morgan Kaufmann, 1999.
- [19] F. Massacci, L. Marraro, *Logical Cryptanalysis as a SAT Problem*, *Journal of Automated Reasoning*, pp. 165 – 203, 24: 165, 2000.
- [20] M. Matsui, *The first experimental cryptanalysis of the Data Encryption Standard*. In Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 1-11. Springer, 1994.
- [21] A. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.
- [22] P. Morawiecki, M. Srebrny, *A SAT-based preimage analysis of reduced Keccak hash functions*. *Inf. Process. Lett.* 113(10-11): 392-397, 2013.
- [23] M. Soos, K. Nohl, and C. Castelluccia, *Extending SAT Solvers to Cryptographic Problems*, *Theory and Applications of Satisfiability Testing - SAT 2009*, 12th International Conference, SAT 2009, Swansea, UK, *Proceedings*, pages 244 – 257, 2009.

Chaos simulation and audio output

Branislav Sobota

Department of Computers
and Informatics

Technical university of Košice
Košice, Slovak Republic
branislav.sobota@tuke.sk

Filip Majcher

Department of Computers
and Informatics

Technical university of Košice
Košice, Slovak Republic
filip.majcher@student.tuke.sk

Martin Sivý

Department of Computers
and Informatics

Technical university of Košice
Košice, Slovak Republic
martin.sivy@tuke.sk

Marián Hudák

Department of Computers
and Informatics

Technical university of Košice
Košice, Slovak Republic
marian.hudak.2@tuke.sk

Abstract— In the paper we present the sound output of analogue signal from the Chua's circuit. We are not interested only in the classic displaying of graphically interesting attractors in the plane, but we are also interested in the sound of them. Changes in Chua's circuit parameters result in a variety of periodic and chaotic attractors that can also be used in music or cryptography. MIDI keys have been used so that a signal with the appropriate tone frequency is generated for each key. The result of using various output signals of the Chua's circuit is presented through the internationally known Slovak folk song "The furnace fell down". Also, we point out the possibilities of using chaos in rehabilitation in hearing testing.

Keywords—chaos, Chua's circuit, attractor, music, MIDI, resonant frequency

I. INTRODUCTION

Development in various areas of human activity requires also visualization. The study of systems generating chaos is a relatively new, dynamically developing area even today. The discovery of the first electronic circuit producing a chaotic signal - the Chua circuit [1] was followed by other electronic circuits generating chaos. This research is not just about analyzing a single system, but also developing other new, unexplored systems of differential equations capable of generating chaos [2] - [8]. The graphical visualization of various chaotic attractors and surfaces associated with the existence of chaos in a given system brings with it a great illustrative visualization of the dynamic changes in the system not only on paper or screen, in 2D or 3D [9] - [18] but also in virtual reality [18] - [22]. Considering the current use of chaos in the following areas: secure communication [23], music [19], [24], navigation and control applications in robotics [25], [26], high-DC-DC converter [27], encryption and random number [28], [29] or rehabilitation [30] is visualization of chaos still relevant. There is also a possibility of using chaos in ICT and virtual-reality systems to help people with disabilities to use technologies.

Another interesting area is not only the answer to the question how chaos sounds in the audio field [19], [24], [31], [32], but whether it can be used for example in diagnosing hearing problems. The chaotic signal depending on the parameters can have a very rich or poor frequency spectrum. By applying the filters [33], the frequency spectrum can be changed or set by the parameters of the Chua's circuit desired frequency spectrum by changing a single parameter [31]. Because some people don't hear just some frequencies, the chaos generator can help generate the song and the test person presses the button if he hears the song. Precisely in hearing testing with the possibility of sending a chaotic signal or a song from the left or right side of the patient, it can help in the diagnosis of the illness of (also) handicapped persons. The possibility of creating a song will be shown on the acquired sounds from the attractors of Chua's circuit using MIDI keys.

This work has been supported by the KEGA grant No. 035TUKE-4/2019: „Virtual-reality technologies and handicapped people education“.

In the next section will be presented Chua's circuit, section III. will point out the possibilities of hearing chaos, section IV. will describe the created application *Audiovisual* and section V. will describe the use of MIDI keys. The knowledge gained in this article will be summarized in the Conclusion.

II. CHUA'S CIRCUIT

In the Chua's circuit (Fig. 1a) there are several parameters that can be modified to change the shape of the CHA. These are linear elements L , C_1 , C_2 and conductance G . Non-linear element NR is a non-linear resistor whose Ampere-Volt characteristic (AVCH) is shown in Fig. 1b and mathematical function $g(u_1)$ - relation (1). By changing the sizes of the segments m_0 , m_1 and m_2 (conductances in mS), it is possible not only to change the symmetrical shape of the AVCH of NR, but also to achieve an asymmetrical AVCH of NR. These are all influences that allow for changes in the shape of CHA, and also the frequency of the state variables over time, which are contained in the system (2) describing the Chua's circuit. We use a time-dependent designation (time symbol "t" in parenthesis) when declaring dynamic (i.e. time-dependent) variables.

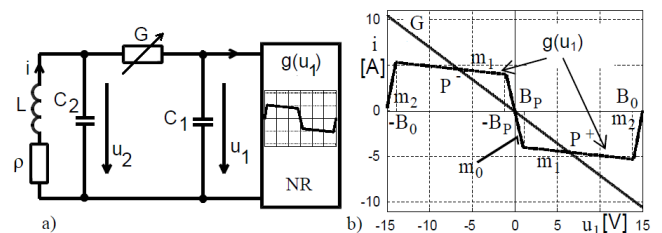


Fig. 1 a) Chua's circuit, b) AVCH of $g(u_1)$.

$$g(u_1) = m_2 u_1 + 0,5(m_1 - m_0)(|u_1 - B_P| - |u_1 + B_P|) + 0,5(m_2 - m_1)(|u_1 - B_0| - |u_1 + B_0|) \quad (1)$$

$$C_1(du_1/dt) = G(u_2(t) - u_1(t)) - g(u_1(t)) \quad (2)$$

$$C_2(du_2/dt) = G(u_1(t) - u_2(t)) + i(t)$$

$$L(di/dt) = -u_2(t) - \rho i(t)$$

where ρ is the coil resistance.

The book [19] contains several hundred CHAs divided into several categories. The first category relates to the Physical Circuit (PC), which corresponds to the circuit of Fig. 1a. In this chapter there are 195 CHAs and we will use the parameters necessary for simulation. Obviously, Chua's circuit at Fig. 1a contains a parallel resonant circuit LC_2 , whose resonant frequency f_0 is given by Thomson's relation

$$f_0 = 1/(2\pi\sqrt{LC_2}) \quad (3)$$

If the circuit generated only periodic oscillations (at $\rho = 0$), their frequency would correspond to the frequency f_0 of (3). Because Chua's circuit generates chaos - it does not have a period - the period refers to the time period T indicated in

Fig. 2a. Also, if $\rho \neq 0$, the use of Thomson's relation (3) is limited. Fig. 2b illustrates a phase portrait of CHA in the plane u_2, u_1 .

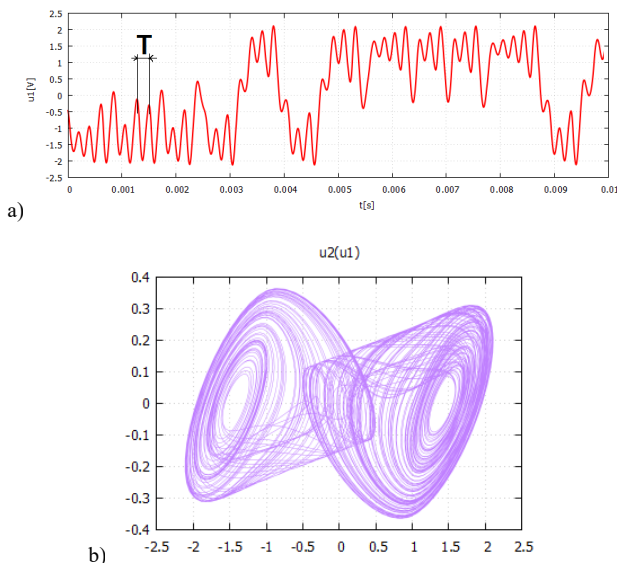


Fig. 2 a) Time waveform $u_1(t)$ and indication of the resonant frequency period f_0 of the parallel LC_2 circuit, for PC1 parameters of [19], b) CHA phase portrait for PC1 parameters.

On the basis of the resonant frequency f_0 , it is possible to quickly calculate whether the output signal e.g. $u_1(t)$ is located in the audible area of the human (20 Hz - 20 kHz). As shown in Tab. 1 for some selected parameters, f_0 is greater than 20 kHz, so that these parameters cannot be used for an audible audio signal. What is needed to be done to listen to any CHA?

III. CHAOS AS AN AUDIBLE AUDIO SIGNAL

There are basically two ways to make CHA audible. The first option has already been used in publication [31]. If the values L , C_1 and C_2 have been multiplied e.g. 10, the frequency f_0 was 10 times smaller than the original. Thus it was possible to select a suitable frequency in the range from 20 Hz to 20 kHz. By using *Matlab*, the frequency spectrums of the signals were generated as well as 7 seconds long *.wav files corresponding to the state variables $u_1(t)$, $u_2(t)$ and $i(t)$. The disadvantage of this method of adjusting the frequency of the signal is that the circuit may not always generate chaos when the values of the elements are changed. The second way is through signal conversion. In the recalculation algorithm, the difference between the original signal frequency and the desired new frequency is first determined. System solution (2) was a constant integration step $t = 5,20833 \cdot 10^{-6}$ s, which corresponds to the maximum sample rate of the sound card used, 192 kHz. If the new frequency f_0 is to be higher, several samples are combined into one using arithmetic mean. If the new frequency is to be lower, more samples are created between the original samples, and the values are calculated as the previous signal sample plus the difference of the next and previous sample of the original signal multiplied by the fraction. This fraction represents how many new samples are between the original samples.

The first possibility of changing the frequency of the signal is suitable if we want to get the signal into the audible frequency range. It maintains the exact shape of the CHA, but tuning it to the exact frequency of the musical tone could be a problem. The second option is suitable just for getting the exact tone. However, it is not suitable for large frequency

variations (e.g. if the original signal had a frequency of 1MHz) because it creates a signal distortion which is negligible in small jumps.

The correctness of the procedure, with unchanged chaotic signal shape after recalculation, was then verified by comparing the created audio file - Fig. 3a and obtained by simulating $u_2(t)$ - Fig. 3b.

In order to calculate, render and sound the selected CHA, an application was developed that made it possible. Its description is in the next chapter. This procedure can also be used in others, e.g. dimensionless systems of differential equations that can generate chaos or are used for analysis of multiple-valued memories where a very long transient event has been observed [34] - [40].

IV. APPLICATION AUDIOVISUAL

In addition to the aforementioned options, this application also enabled export from another application that used to calculate the trajectories and boundaries of the Chua's circuit using CPU parallelization [41]. The flow diagram is shown in Fig. 4. This illustrates how *Audiovisual* works, which will be described with the GUI shown in Fig. 5.

Since the *Audiovisual* application is created in Slovak, the individual options are indicated by numbers 1 - 6. In option 1, the Input signal type can be selected. This can be calculated from [41], or choose the second option, which is selected in Fig. 6 - Circuit parameters. In option 2 it is necessary to enter the name of the input file (our choice is PC1P) and we also choose one of the options of rendering the graphic output (1 or 3 windows), Create a gallery, Transform the signal to another frequency or just create an audio signal. In option 3, one of the state variables has to be selected and plotted in time $i(t)$, $u_2(t)$, or $u_1(t)$. In the fourth option we select from symmetric or asymmetric AVCH with the possibility of its display. Option 5 changes the thickness of the curve (trajectory) to be plotted and in option 6 we enter the desired frequency. An example of the output of the Audiovisual application is shown in Fig. 7. *Gnuplot* was used to draw in each window. We were interested not only in how chaos sounds, but also in the possibility of loading samples of a chaotic signal through the MIDI keyboard and using it to play a song. If the signal was non-periodic i.e. chaotic sounded like white noise. It can also be used in music, it may have a higher or lower (carrier) frequency, but it can not be considered a specific tone. Some output signals were periodic, they

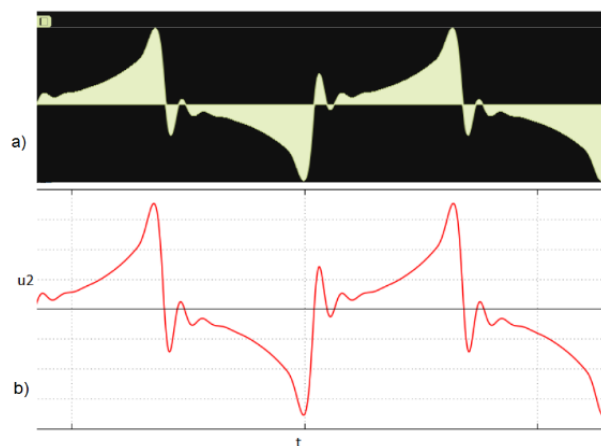


Fig. 3 Comparison of part of the chaotic signal a) audio file, b) $u_2(t)$ from the simulation corresponding to the PC25 parameters.



Fig. 5 Audiovisual flow control diagram

sounded as a tone of a certain frequency. By fine-tuning the physical parameters of the circuit, we were able to obtain a multi-loop signal (Fig. 8a), the more they were, the more interesting the tone was. An example of $u_2(t)$ is illustrated in Fig. 8b. Different CHAs have different shapes, so the color tones of different CHAs are different and therefore it was interesting how chaos could be used when playing songs on MIDI keys. The aim of the experiment was not to study human perception of sound, but to create an interesting and audible audio signal.

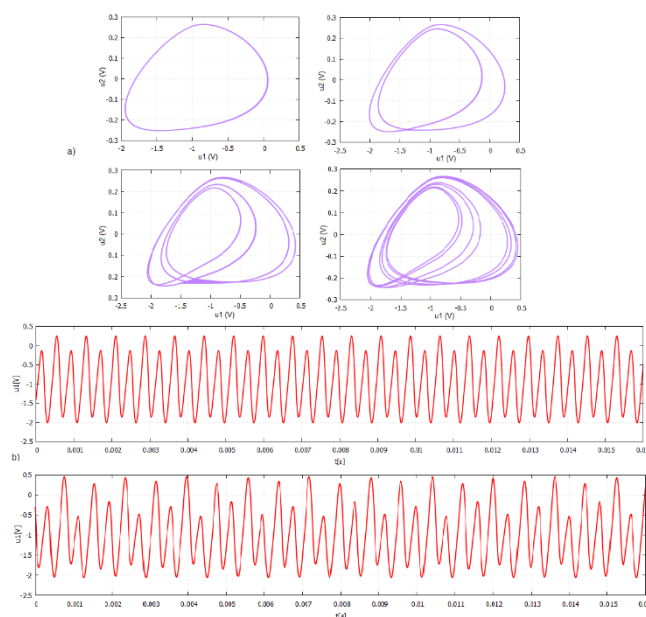


Fig. 7 a) Phase portraits of the periodic signals obtained by adjusting the parameters - from left to right: one, two, four and eight loops, b) time course $u_1(t)$ for two respectively eight loop stage portrait.

TABLE I. EXAMPLE OF CALCULATED RESONANT FREQUENCY f_0 OF PARALLEL LC_2 FOR SOME SELECTED PC PARAMETERS FROM [19].

Parameters	C_1 (nF)	C_2 (nF)	G (mS)	L (mH)	ρ (Ω)	m_0 (mS)	m_1 (mS)	f_0 (Hz)
PC1	0.10443	1	1.011	0.0625	0	-1.143	-0.714	636.619
PC3	0.10443	0.85	1	0.0625	0	-1.143	-0.714	690.511
PC7	-0.1333	9.7	1	0.31	-0.1	-0.98	-2.4	91.781
PC16	0.06	11.65	-1	0.00667	0.000651	0.856	1.1	570.945
PC25	1	-1.12	33.3333	-1.49	2.228	-0.5	0.0064	123.202
PC50	0.26	100	-0.699986	0.136	0.000929	0.599	0.77	43.156
PC66	1	-1.056	18.867925	-12	2.228	-100	-0.003	44.709
PC80	1	-1.0837	33.932813	-8.824	4.65	-0.5	-0.0064	51.467
PC133	0.58	10	-1	0.047	0.0041	-0.474	2.039	232.151
PC194	-0.1333	10	1	0.237	-0.1	-0.98	-2.4	895.428

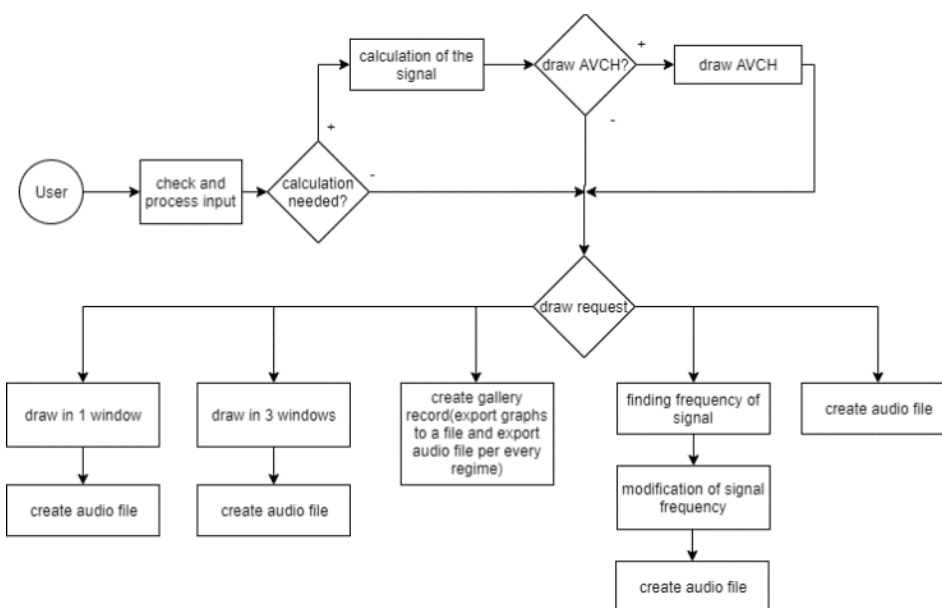


Fig. 4 Audiovisual flow control diagram.

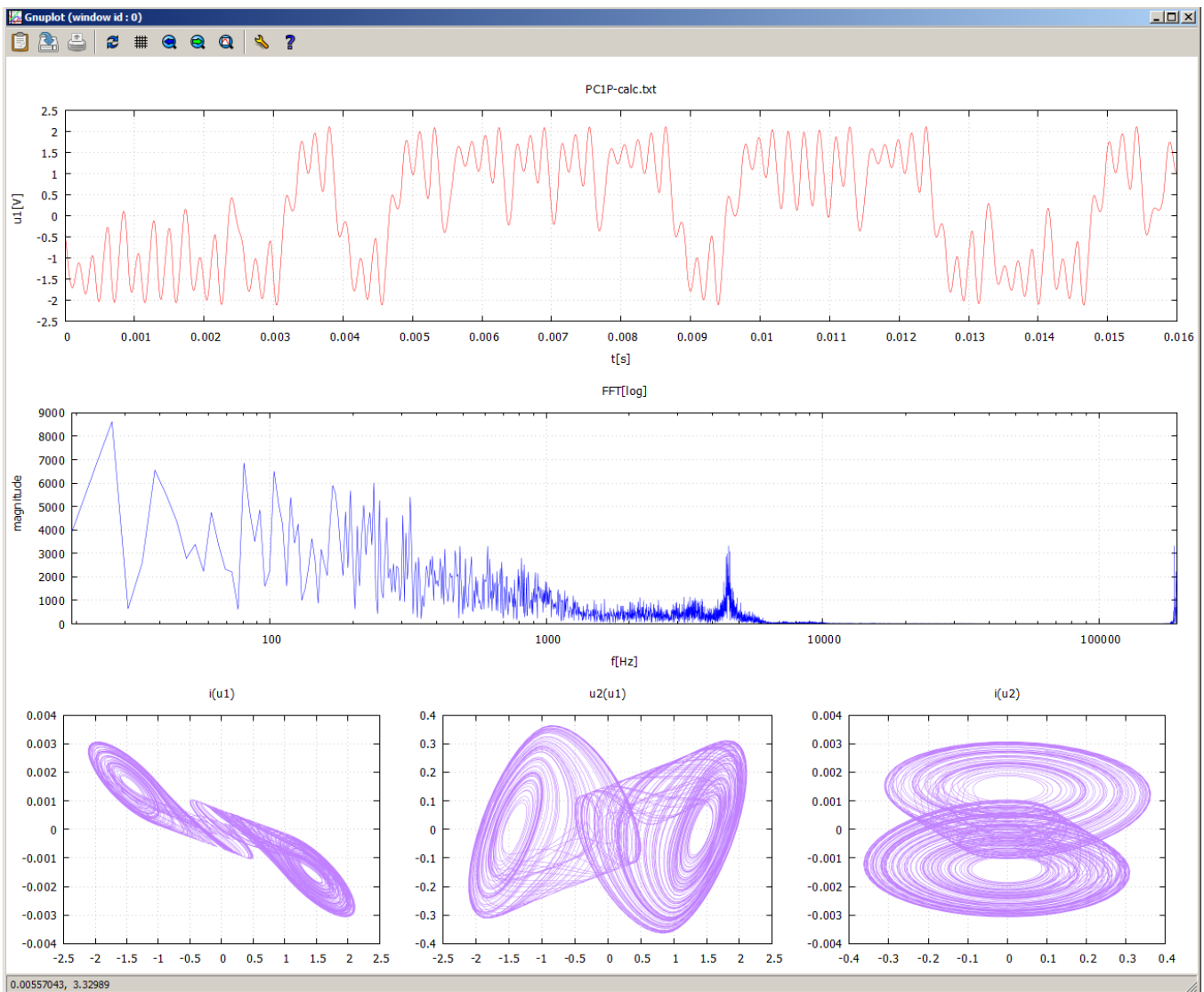


Fig. 6 Graphical output of *Audiovisual*. Up is the time course $u_1(t)$, in the middle is the frequency response $u_1(t)$ and below are three phase portraits of the CHA of the system (2).

V. USE OF MIDI KEYS

The ability to import an *.wav file into MIDI keys created by *Audiovisual* is twofold. The first option is that a function will be implemented to alternate the frequency of the input signal, and with it it is possible to create a signal for each key with the frequency of the respective tone. The second way would be to leave their tones with the key and "mix" the chaotic signal (*.wav) generated to them. The user could select the volume of the added sound using the "gain" setting.

Both of these methods can be applied at the same time, so that each key has its own tone in addition to its own chaotic sound, which will be added to it.

We applied the first option. *Vmini* MIDI keys from *Alesis* were used. They include a dual octave keyboard, 4 adjustable pads, octave change buttons and many other features. Free *MuLab* software was used. From one sample, *MuLab* can create a tone of adequate frequency for the entire keyboard. Thus all 195 attractors from [19] were modified and imported. As a test song we decided to choose the Slovak national and internationally known song "*Pec nám spadla*" (*The furnace fell down*). We prepare also next 20 different sound. They are available as about 5 seconds .wav sample. For the file

designation was used name PCXYZPEC.wav, where PC means Physical Circuit, XYZ serial number of the parameters listed in [19].

VI. CONCLUSION

In the article was presented possibility of hearing the chaos and also (it was the main motivation) the use of interesting sounding attractors of Chua's Circuit when playing on MIDI keys. The application was created in C language and GUI in C++. Its graphic output is suitable for monitoring the time course of the output, frequency spectrum and also in the form of phase portraits of individual attractors. After listening to 195 attractors, 20 samples were selected for this article, which were provided to the MIDI keyboard. Tones of adequate frequency have been calculated for the entire keyboard and some melody could be played. The internationally known Slovak folk song "The furnace fell down", has been selected, and its about 5 seconds long preview is on the website. We leave choosing of the most interesting sub-color of the song melody to the reader of this contribution. This CHA "audio visualization" procedure is also possible to use for dimensionless chaos-generating systems, or for systems of differential equations, whose solution is characterized by a very long transition process, where change of tone in time may

be even more interesting. Another important possibility is the use of the spectrum of frequencies characteristic for CHA in rehabilitation in hearing testing.

REFERENCES

- [1] T. Matsumoto, "A chaotic attractor from Chua's circuit," *IEEE Transactions on Circuits & Systems*, vol. CAS-31., no. 12, pp. 1055-1058, Dec. 1984.
- [2] T. Yang and L. O. Chua, "Piecewise-linear chaotic systems with a single equilibrium point," *International Journal of Bifurcation and Chaos*, vol. 10, no. 9, pp. 2015-2060, 2000.
- [3] Q. Lai, P. D. K. Kuate, F. Liu and H. H. Ch. Iu, "An extremely simple chaotic system with infinitely many coexisting attractors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1-4, 2019.
- [4] V. I. Vagaitsev, N. V. Kuznetsov and G. A. Leonov, "Localization of hidden attractors of the generalized Chua system based on the method of harmonic balance," *Vestnik St. Petersburg University. Mathematics*, 2010, vol. 43, no. 4, pp. 242-255, 2010.
- [5] J. Petržela and M. Guzan, "Analog implementations of dynamical systems with equilibria degenerated into plane objects," *39th International Conference on Telecommunications and Signal Processing (TSP)*, Vienna, Austria, jun 27-29, pp. 243-249, 2016.
- [6] T. Gotthans, J. C. Sprott and J. Petržela, "Simple chaotic flow with circle and square equilibrium," *International Journal of Bifurcation and Chaos*, vol. 26, no. 08, pp. 1650137-1650147, 2016.
- [7] G. Xu, Y. Shekofteh, A. Akgül, Ch. Li and S. Panahi, "A new chaotic system with a self-excited attractor: Entropy measurement, signal encryption, and parameter estimation," *Entropy*, 20, 86, pp. 1-23, 2018.
- [8] J. Petržela, "Strange attractors generated by multiple-valued static memory cell with polynomial approximation of resonant tunneling diodes," *Entropy*, vol. 20, no. 9, p. 1-23, 2018.
- [9] T. Matsumoto, L. Chua and M. Komuro, "The double scroll," *IEEE Transactions on Circuits and Systems*, vol. 32, no. 8, pp. 797-818, 1985.
- [10] V. Špány and L. Pivka, "Boundary surfaces in sequential circuits," *International Journal of Circuit Theory and Applications*, vol. 18, no. 4, pp. 349-360, 1990.
- [11] M. Guzan, and B. Sobota, "3D visualization of Chaos in state space," *Electrical Drives and Power Electronics (EDPE)*, High Tatras, Slovakia, sept. 21-23, pp. 130-136, 2015.
- [12] J. Petržela, "Multi-state memory Cell with Resonant Tunneling Diodes: Circuit Tool for Chaos Generation," *28th International Conference Radioelektronika, Prague*, pp. 1-5, 2018.
- [13] M. Guzan, "Variations of boundary surface in Chua's circuit," *Radioengineering*, vol. 24, no. 3, pp. 814-823, 2015.
- [14] L.J. Ontañón-García and E. Campos-Cantón, "Widening of the basins of attraction of a multistable switching dynamical system with the location of symmetric equilibria," *Nonlinear Analysis: Hybrid Systems*, vol. 26, pp. 38-47, 2017.
- [15] M. Guzan, P. Kováč, I. Kováčová, et al., "Boundary surface of Chua's circuit in 3D state space," *International Conference on Modern Electrical and Energy Systems (MEES)*, Kremenchuk, Ukraine, nov. 15-17, pp. 388-391, 2017.
- [16] Q. Lai, A. Akgul, Ch. Li, G. Xu and U. Cavusoglu, "A new chaotic system with multiple attractors: Dynamic analysis, circuit realization and S-box design," *Entropy*, vol. 21, no. 12, p. 1-15, 2018.
- [17] J. Petržela, T. Gotthans and M. Guzan, "Current-mode network structures dedicated for simulation of dynamical systems with plane continuum of equilibrium," *Journal of Circuits Systems and Computers*, vol. 27, no. 9, pp. 1-39, 2018.
- [18] B. Sobota and M. Guzan, "Virtualization of Chua's circuit state space," *Recent Advances in chaotic systems and synchronization: From theory to real world applications*, Book Series: Emerging Methodologies and Applications in Modelling, Identification and Control, Elsevier, pp. 127-163, 2019.
- [19] E. Bilotta and P. Pantano, "A gallery of Chua attractors," *World Scientific, Series A - vol. 61*, pp. 607, 2008.
- [20] B. Sobota and M. Guzan, "Possibilities of utilization chaos for the cognitive tests using virtual reality technologies," *16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Stary Smokovec, Slovakia, nov. 15-16, pp. 499-504, 2018.
- [21] Chaos theory VR, [online] Available: <http://www.chaostheoryvr.com/>
- [22] Virtual reality: Chaos theory, [online] Available: <https://www.royalroad.com/fiction/23424/virtual-reality-chaos-theory>
- [23] G. Kaddoum, "Wireless chaos-based communication systems: A comprehensive survey," *IEEE Access*, vol. 4, pp. 2621-2648, 2016.
- [24] W. Wainiya and P. Sooraksa, "Chen's attractor in music composition," *21st International Computer Science and Engineering Conference*, pp. 93-96, 2017.
- [25] A. Buscarino, L. Fortuna, M. Frasca, and G. Muscato, "Chaos does help motion control," *Int. J. Bifurcation Chaos*, vol. 17, no. 10, pp. 3577-3581, 2007.
- [26] Q. Zhang, M. Yaan and R. Song, "Robot trajectory planning method based on genetic chaos optimization algorithm," *18th International Conference on Advanced Robotics*, pp. 602-607, 2017.
- [27] A. Ragusa and G. Marsala, "EMI reduction in low input ripple DC-DC converter using chaos PWM technique," *Journal of Electronic Research and Application*, vol. 2, no 2, pp. 42-47, 2018.
- [28] M. Drutarovský and P. Galajda, "A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware," *Radioengineering*, vol. 16, no. 3, pp. 120-127, 2007.
- [29] S. Ergun, "Vulnerability analysis of a chaos-based random number Generator," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3331-3334, 2018.
- [30] D. Šimšík, A. Galajdová, M. Drutarovský, P. Galajda and P. Pavlov, "Wearable non-invasive computer controlled system for improving of seniors gait," *International Journal of Rehabilitation Research*, vol. 32, no. 1, p. 35-38, 2009.
- [31] M. Guzan, M. Olejár, V. Cviklovič and J. Petržela, "Simulation of the Chua's circuit with audio output," *26th International Conference Radioelektronika, Košice, Slovakia*, pp. 1-4, 2016.
- [32] G. Mayer-Kress, I. Choi, N. Weber et al., "Musical signals from Chua's circuit," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 688-695, 1993.
- [33] J. Petržela, R. Sotner and M. Guzan, "Implementation of constant phase elements using low-Q band-pass and band-reject filtering sections," *21st International Conference on Applied Electronics (AE)*, Pilsen, Czech republic, sept. 06-07, pp. 205-209, 2016.
- [34] M. Guzan, "Boundary surface and stable manifold in sequential circuits," *21st International Conference on Radioelektronika, Brno, Czech republic*, pp. 219-222, 2011.
- [35] P. Galajda, M. Guzan and V. Špány, "The control of a memory cell with the multiple stable states. 21st International Conference on Radioelektronika, Brno, Czech republic, p. 211-214, 2011.
- [36] M. Stork, "Coexistence of multiple chaotic attractors," *28th International Conference Radioelektronika, Prague, Czech Republic*, April 19 - 20, 2018, pp.1-5, 2018.
- [37] M. Guzan, "Boundary surface of a ternary memory in the absence of limit cycles," *22 st International Conference on Radioelektronika 2012. Brno, Czech republic*, pp. 1-4, 2012..
- [38] J. Petržela, "Three-segment piecewise-linear vector fields with orthogonal eigenspaces," *Acta Electrotechnica et Informatica*, vol. 9, no. 2, pp. 44-50, 2009.
- [39] J. Petržela, "Multi-valued static memory with resonant tunneling diodes as natural source of chaos," *Nonlinear Dyn (2018)* 94, pp. 1867-1887.
- [40] M. Guzan, "Boundary surface of 5-valued memory," *Journal of Engineering*, vol. 2013, pp. 1-7, 2013.
- [41] Z. Racz, B. Sobota and M. Guzan "Parallelizing boundary surface computation of Chua's circuit," *Radioelektronika 2017, Brno, Czech republic*, pp. 1-4, 2017.

Some innovations of teaching the course on Data structures and algorithms

1st William Steingartner
*Faculty of Electrical Engineering
and Informatics*
Technical University of Košice
Košice, Slovakia
william.steingartner@tuke.sk

2nd Ján Eged
*Faculty of Electrical Engineering
and Informatics*
Technical University of Košice
Košice, Slovakia
jan.eged@student.tuke.sk

3rd Davorka Radaković
Faculty of Sciences
University of Novi Sad
Novi Sad, Serbia
davorkar@dmi.uns.ac.rs

4th Valerie Novitzká
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
valerie.novitzka@tuke.sk

Abstract—The teaching of practical foundations of computer science is still a big challenge. The course on Data Structures and Algorithms is one of the most important foundational courses that are necessary to be included in the curriculum of future IT experts. The content of this course is quite wide and it differs in many universities. In our teaching process, we identified some interesting problems that could be presented more attractively – using the visualizing software. In this paper, we present a purpose, rôle, design and methodology of our developed software tool for the convenience of teaching and studying the problem of the longest common subsequence.

Index Terms—data structures and algorithms, formal methods, software products, software engineering, teaching innovation, teaching software, university didactic

I. INTRODUCTION

The course on Data Structures and Algorithms is a standard part of a curriculum for a computer science study [34]–[37]. Within this course, many traditional algorithms for searching, sorting or manipulating with various data structures are presented. To teach algorithms is nowadays a great challenge: using modern technologies can simplify the teaching process and to reduce the time necessary for explaining the principles. Modern technologies cover mostly the visualizing software that helps to understand the basic principles and to see particular steps of an algorithm when applied to concrete values.

The longest common subsequence (LCS) problem is the problem of finding the longest subsequence common to all sequences in a set of sequences (often just two sequences). It is a classic computer science problem, the basis of data comparison programs such as the *diff* utility, and has applications in computational linguistics and bioinformatics.

This work has been supported by (1) the Slovak Research and Development Agency under the contract No. SK-AT-2017-0012: Semantics technologies for computer science education; and by (2) the project KEGA 002 TUKE- 4/2017: Innovative didactic methods in education at university and their importance in increasing education mastership of teachers and development of students' competences.

A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements (the order should be from left-to-right, not necessarily consecutive). Longest common subsequence (LCS) of two sequences is a subsequence, with maximal length, which is common to both the sequences.

For the given two sequences of integers $A = (a_1, \dots, a_n)$, and $B = (b_1, \dots, b_m)$, the goal is to find the longest common subsequence and to print it.

Finding the longest common subsequence has applications in areas like biology. The longest subsequence (LCS) problem has an optimal substructure property. Thus, the dynamic programming method can be used to solve this problem.

The paper is organized as follows: Section II presents motivation for the development of a learning tool package for teaching faculty subject Data structures and algorithms. Section III provides an overview of related work. The user interface design, system design and database design is shown in Section IV. Section IV presents three most important parts: the database, UI and algorithms used in this tool. Real life use of this tool is given in Section VI. Section VII concludes the paper.

II. MOTIVATION

The main reason behind the development of this application was to create a useful tool to work with various number sequences and their subsequences. It is expected to be a part of the learning tools package for a subject Data structures and algorithms. The use of dynamic programming and effective algorithms was necessary, to make the application run smoothly and efficiently.

The application includes the usage of several algorithms to search for subsequences, such as longest increasing and decreasing subsequence, longest consecutive increasing and decreasing subsequence or longest common increasing and decreasing subsequence. All of these algorithms were developed,

to be as quick as possible, using the dynamic programming method, specifically for number sequences. There are other methods, for instance, divide and conquer or brute force algorithms, which we did not specialize on, but are sometimes very effective as well. The plan was, to create an executable JAR file, using object-oriented programming language Java, which could be launched independently of any other components. Also, we decided, that to store data from the use of the processed sequence, there would be two options.

The first option to save the data includes saving data logs via file explorer. For example, if an algorithm was used to process a sequence, the data log would be created afterwards and would appear in the area of results. The user could then click the save button, which would open the File Explorer and give the user an option to type the name and select the destination path for the saved file. For the second option, attaching a simple local database to the application is needed. That means, that it is required to have the database installed. The advantage of the H2 database is, that it's system requirements are minimal. Also, as it doesn't require too much space, it is very quick and stable, which is essential for the smooth run of the application itself. After running one of the algorithms, an alert would appear asking, if the user would like to save the results to the local database as well. The table of the saved data logs would appear after clicking the database button. Only the data logs and results of a logged-in user would be visible, to accomplish some level of privacy. With the help of the database, users would be able to register, login and log out as well.

Due to the fact, that this application's main purpose is to help students try and experiment with solving tasks connected to number sequences, it would be redundant to develop a web application. Instead, we oriented on the development of a desktop application, involving a simple local database, that would be easy to run regardless of any server or internet connection problems. This way, the application would be able to run properly under any circumstances, with the benefit of quick loading and easy manipulation. A very important part of every application is its user interface. We chose to use Swing and some parts of abstract window toolkit, such as alerts, warnings and so on. Swing provides various useful components, like buttons, text areas, labels, frames or tooltips so that the application could end up being very complex and understandable.

III. RELATED WORK

One of the most fundamental problems in computer science is the string similarity measure. The string edit distance belongs to a family of metrics that reflect the number of operations needed to transform a string A into a string B [15]. To measure that, a three distance metrics are proposed: Hamming distance [11], Longest Common Subsequence (LCS) distance [2], [19], and Levenshtein distance (also known as edit distance) [22], [30]. The Longest Common Subsequence is the longest possibly non-consecutive subsequence that is common to both strings. Longest Common Subsequence distance is one

of the most commonly used distance function which solves a problem of string matching allowing errors, i.e. approximate string matching [2], [4], [19], [21]. This problem was firstly studied in the 1960s and 1970s and it still interesting for studying and improving time complexity [6], [8], [10]. It covers big application areas such as: computational biology of DNA, RNA and protein sequences, storing, copying, and processing digital information encoded in the letters A , C , G , and $T(U)$ [1], [14], [18], [25], [32], [38], [39]; file comparison [3]; signal processing, i.e. speech recognition [20]; information retrieval (IR) - solving problem of correcting misspelled words in written text [7], [24]; plagiarism detection [5]; handwriting recognition [16]; image compression; data mining [29]; pattern recognition [19], etc.

Wagner and Fischer in 1974 proposed the algorithm for the string-to-string correction problem, which determines the distance between two strings measured by the minimum cost sequence of "edit operations" needed to change the one string into the other in time proportional to the product of the lengths of the two strings [33]. Hirschberg in 1975 presented an improved algorithm which solves this problem in quadratic time and linear space [12]. Next, Hunt and Szymanski proposed an algorithm which has a running time of $O((r+n)\log n)$, where r is the total number of ordered pairs of positions at which the two sequences match [17]. Correspondingly, Hirschberg proposed $O(pn)$ algorithm and $O(p(m-p)\log n)$ algorithm, where p is the length of an LCS and m and n are the lengths of given two strings ($m \leq n$). [13]. In 1984, Hsu and Du gave better algorithm with $O(pm \log(nm) + pm)$ [15].

IV. DESIGN METHODOLOGY

This section deals with user interface design, system design and database design. It orients on conceptual model, screen sequence design, iterative development, results and responses from prototype designing, hierarchical task analysis diagram, use case diagram and relational scheme of of the used database. Interactions and relations between objects and components within the design are described in detail both generally and on practical examples as well.

A. User interface design

Within the software design, no matter if it is dedicated to Android, Windows, Linux, or other platforms, it is always needed, to think about the user interface. Users communicate and work with the application through user interface, so the design of the user interface is essential. Professional user interface should be intuitive, maintainable, simple and attractive. To reach all these qualities, the design shouldn't be neglected and various helpful tools should be used for the specification of the interface and suitable manipulation with it.

The conceptual model consists of several main items, represented by frames. These items are User, Input, Output, Detailed output, Algorithms and Account. There are also minor items, such as Database, File, Manual output, Longest common increasing / decreasing subsequence and others. User

is the central item, which in conclusion means, that he is the one to perform the actions. Name of each action is shown near the arrow. Arrows represent those actions. The User item is connected to the other main items directly, which means, that he can perform or execute the actions represented by arrows on all those objects. Some objects can also have more types. The example of such an object is object Input. Unfilled arrow references to the descendants, that inherit from the parent object, which is Input. The descendants of the object Input are File and Manual input. On the same principle, Database and File are child objects to main object Output. No other object apart from user can execute actions, so all the actions performed on these objects, are caused by an impulse from the user.

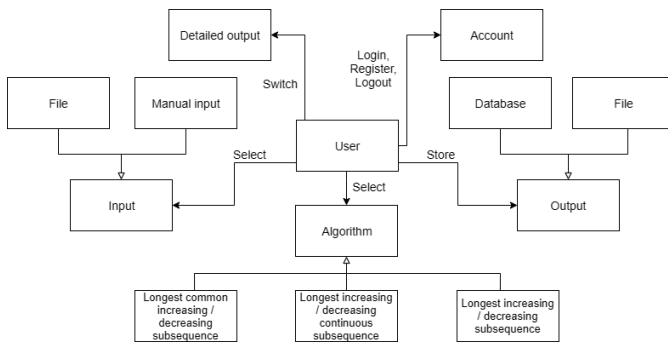


Fig. 1. Conceptual model

Based on the knowledge above, the user can perform actions, such as save the output generated by the program into a file, with the help of file Explorer, or save the results directly to the database, choose the way to enter input, either manually or using a file, user can also select the algorithm to process one or more number sequences, switch to detailed mode, register his account, log in to an existing account, logout as well and so on. For the creation of the conceptual model, an internet tool draw.io was used. It is intended for production of UML, ER, network and many other useful diagrams and schemes.

The screen sequence diagram consists of frames, representing screens and actions. Each of the main screens differs from the actions, located on a certain screen, by color indication. For instance, there is an arrow beginning on the Home screen and it enters the Input processing, which would normally be a dead end. However, it applies that Input processing screen is just an action, that can be executed within the Home screen. The Home screen represents the main screen of the application, that is first visible after running the application. The other two screens, Log in / Log out and Database screen only appear after certain impulse, carried out on the Home screen. Action names are written near the arrows and those arrows represent the impulses. The arrow orientation shows, which element initiates the action. To clear things about the functionality up, action can be performed like this: The user runs the application and is shown the Home screen. Afterwards, he navigates the cursor to the bar on the upper part of the application, selects Account and Log in option. The Log in / Registration screen

show up and the user can enter his username and password. If the entered data is correct, the Log in / Registration frame will close and his username will appear in the Logged in user label on the main screen. For the creation of the screen sequence diagram, same web tool was used, as for the conceptual model, draw.io.

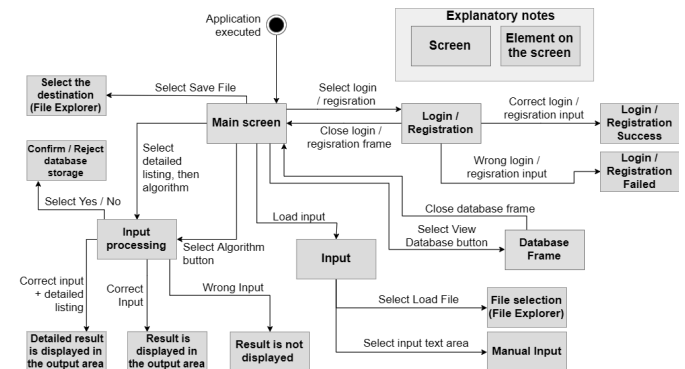


Fig. 2. Screen sequence diagram

The whole design of the user interface was created iteratively, with the orientation on user, in two different prototypes. The first prototype was created with prototyping tool Balsamiq Mockups 3 and verified through the first user testing. The second prototype was created with much more complex and harder to operate tool, Justinmind Prototyper 8.6.1. The user testing was performed with two different user groups, consisting of three people, in each iteration. The users reviewed both of the prototypes in terms of usability, attractiveness, intuitiveness, and easy learnability of use, in order to collect as much useful information as possible, before creating the final version of the application. For the first testing, Balsamiq Mockups 3 was selected for a few reasons. It is quite simple to use for beginners and considering that the first prototype should only be as much as a foundation for the application, developing during each iteration, it made this tool an ideal option. Using it, it is possible to create transitions based on hyperlinks, while lower level functionality may be added as well. The disadvantage of this prototyping tool is that it might be too simple, as there is no option to create animations and more complicated functionality. Based on this information, it was certain, that even though this tool served well for the first prototype, it was not a suitable candidate for the second, more complex prototype. With the help of the second prototyping tool, Justinmind Prototyper 8.6.1, it was suddenly possible to create more functionalities. The reason is, that this tool is significantly more complex and it contains much more functionality possibilities and utilities. Shortcomings from the previous prototype were solved, such as button mapping, more realistic design and responses to user input. The users testing the second prototype had only problems with the lack of saved data, leading to incorrect results of certain actions, however, this problem was solved with storing the real data, later during the implementation. The first testing showed both major and minor flaws. Users came up with several

suggestions, for instance improvement of the buttons mapping, aesthetics, or graphical interface orientation. Even though the flaws, the general impression was not bad at all. According to the users, the biggest disadvantage was lack of functionality testing, which the prototype was supposed to support based on scenarios. Right after the verification, the deficiencies were fixed. However, the testing of the second prototype revealed a radical progress, as the testers were very satisfied with the interface and design of the software, as they were able to navigate intuitively and the application was just a few steps away from the final product. The prototype wasn't perfect yet, but it was a big step forward. The testing also showed, that even though Balsamiq Mockups is really simple and easy to use, it doesn't match the professional JustInMind tool, full of almost endless possibilities.

B. System design

The system design is necessary in terms of functionality of the application [31]. With the help of hierarchical task analysis diagram and use case diagram, functions, information about each step and results of the actions were explained briefly and clearly. Two of the most important domain-specific actions were explained in the HTA (Hierarchical Task Analysis) diagram. The first action is entering the system or manipulation with the system after launch. There are three fundamental options, which are registration, log in and guest mode. In case of registration, unique username and password are needed. If the username doesn't match any other name in the database and the characters are compatible with the system, the data is saved to the database and user can then log in. User can successfully log in using the correct combination of the username and password and can then use the application as a logged in user. By choosing not to register or log in, user is considered a guest. All the guest users share their own database with all of their saved results and data logs. The second action defines the main purpose of the application. This functionality is much more complex and is focused on searching for the subsequence in various steps. Firstly, input data is required. User can choose one or more subsequences, depending on the algorithm. There are also two options for entering the number sequences (manual input and file content). After that, one of the algorithms needs to be picked, whether it needs two, or one input sequence, all incorrect combinations are dealt with.

During the system design, use case diagram is usually not skipped. It covers the specification of expected behaviour of the software. That means, that it doesn't explain specific methods, with which the functionality is addressed. It rather explains what objectives the system achieve. The diagram contains an actor, who works with the system. The actor starts the communication with the system and is not reactive, which means, he is the primary actor. There can be more actors, but it depends on the design of a specific system. The system itself is represented with a frame, where all of the content, except actor, is located. Use cases are illustrated with named oval shapes. The diagram also contains some lines and arrows, that represent relations. Full lines, that connect the user with

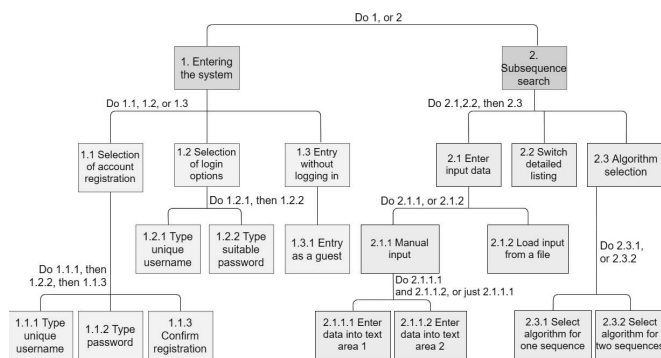


Fig. 3. Hierarchical task analysis diagram

the use cases, are called associations. Association represents the main interaction. Dashed line, with the text `<<Include>>` and arrow pointing to the minor case, represents the included use case and it means the action is mandatory. For example, in the algorithm choice use case, the verification of correct input is always included. However, a dashed line with the text `<<Extend>>` and arrow pointing to the primary case, means, that the minor action only occurs under certain circumstances. For instance, the output manipulation use case does not have to involve saving of the data log. Lastly, the full line with the unfilled arrow, pointing to a primary use case represents child objects.

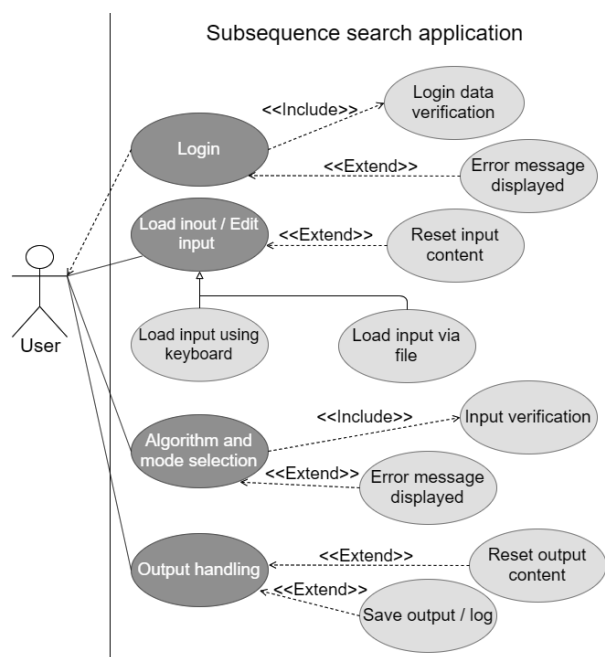


Fig. 4. Use case diagram

C. Database design

The database system serves as a reliable preservation of the saved data, with which an further use may be required. Additional database storage solution was preferred over just

simple file storage, as it is considered the most secure way of data storage. The database was represented by a relational scheme. It consists of two main parts, sequence logs table and registered user table. The sequence log table has nine attributes, which are username, id, algorithm, first input sequence, second input sequence, output sequence, output length, algorithm length and time and date of the creation. The second table serves to store data about registered users. It contains four attributes of which the first is unique user name, serving as a primary key. Then there is user password, which is essential for log-in action, id and date and time of the creation of certain account.

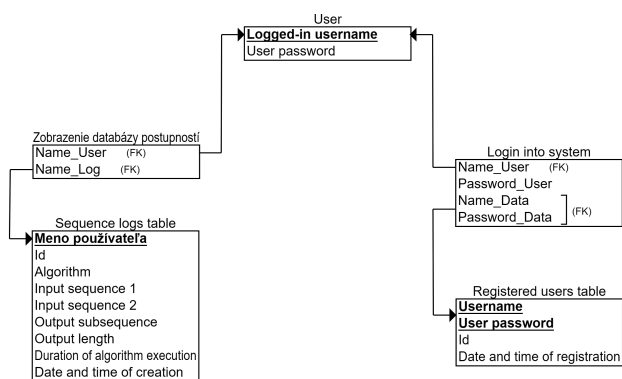


Fig. 5. Relational schema of the database

V. APPLICATION COMPONENTS

There are three most important parts of the application. These are the database, user interface and algorithms themselves. Every one of them has its own role in case of the whole system and are essential parts of the application

The first step to develop an application used for processing number sequences, based on chosen algorithms, was of course, assembling methods for all of the used algorithms. To achieve this goal, several methods could be used. For example, a String type method, an `int[]` type method, however, the most suitable option would be `List < Integer >`. Compared to integer arrays, it has an advantage, that it is much more simple to manipulate with. In addition, in case of dynamic data, which do not have a fixed size, the `List` is far more effective. For all of the algorithms, a dynamic programming solution was applied, as it was an essential requirement.

The algorithms were observed under the same conditions and circumstances, numerous times and with various numeric sequences. Plenty of algorithm variations were used to search for a certain subsequence and its length. It was then easier to analyze, which of the variations behave responsibly most of the time and should be used in the application. The result of the comparisons achieved selection of the most effective algorithm in the final version of the application. The main goal, to reach the highest optimality possible and lower the memory and time consumption, was achieved through further testing.

The results of the algorithms showed, that the algorithm to search for longest increasing/decreasing subsequences is the most time consuming one, even though it only operates with one entered numeric sequence. The main reason is, that it creates and stores large amounts of suitable subsequences, out of which it chooses the longest variant. The algorithm to find the longest increasing/decreasing continuous subsequence is the quickest of them all. It is due to the fact, that this algorithm only adds the neighboring element into the suitable subsequence and only works with one input sequence. Searching for the longest common increasing/decreasing subsequence requires two entered sequences and it checks, either there is a common subsequence between the two or not. Compared to the first, similar algorithm, it skips some of the steps, due to searching for match. This means that it is not storing loads of subsequences, hence it is slightly more efficient.

The user interface was created using Swing and few functionalities from AWT. The swing components are written in java, so the platform independence is granted. The main class extends `JFrame` class and implements `ActionListener` interface. The resolution of the main frame is set to 1024×768 , as no bigger resolution is needed for the component layout. Using swing, various components such as buttons, text areas, menu bars, labels, scrollpanes, panels and more important components are created. For the components layout, `GridBagLayout` was used and for the component size and positioning, a `GridBagConstraints` type variable was added as well, for the sake of axis x position, width and so on. `ActionListeners` are added to the components, to connect them to the functions.

Most of the UI components are located on the Main frame. Its is the place where most of the application functionality is situated. The labels are very simple and informative, keeping the application simple and easy to use and orient in. The main menu bar is on the top of the main frame. It contains two menus, File and Account.

File menu consists of three items. Selecting the first item, Open, opens the file explorer and lets the user load a numeric sequence from a certain file into the input text area. The second item, Save, opens the file explorer as well and user can then choose a file name and a path to save the data from the output text area into a file. The last item of the File menu is Exit and it opens a confirmation dialog window and quits the application.

The Account menu offers three options as well, Login, Register and Logout. Login and Register items open the Login / Registration frame, which is the second of the three involved. It is a secondary frame, as it only involves two possible actions. By entering the valid data into text fields, user can login to an existing account or register a new account. Selecting the confirmation button triggers input check. If the entered input is incorrect, username or passwords are invalid or the input length is inappropriate, an error dialog appears. If the login is successful, user's name appears at the bottom right corner. The logout item switches the bottom right label back to default user.

There are three text areas displayed on the Main frame, while two of them are editable and serve as manual input for

a number sequences. The third one displays results after the execution of the program.

Users can also load and save files via the Open and Save File buttons. The reset button resets the data in all of the text areas on the Main frame. Toggling the Detailed checkbox leads to more detailed information in the Results text area. The algorithm buttons are labeled and distinguished from the other buttons by color. Selecting any of these buttons triggers the main function of the application. Firstly, the input gets formatted and checked, whether it's valid. Then, based on the button, the algorithm executes on the input sequence and creates the appropriate subsequence, with all the important data shown in the result Text area. Lastly, an option dialog asks the user, either he wants to store the data into the database or not. The Open database button is very straightforward, as it opens the last of the three frames, Database frame.

The Database frame represents the data stored in the H2 database by a currently logged in user. If no one is logged in at the time, the frame displays default user's data. This frame consists of labels and text fields, creating sort of a table, which is very easy to understand and navigate in. All the important information is displayed there.

All the incorrect input and manipulation within the application is followed by either error or option dialogs, to make the program understandable. Labels and tooltips make it clear, what everything means and what is required from a user. Scroll panes are added to the text fields and areas to make them more practical to look through and resizing of the frames is disabled, to avoid the UI glitches.

For a safe way to store the data, a communication with the H2 database was implemented in addition to file storage option. As already mentioned, the first table serves to store data about the registered users and the second table stores the saved sequence logs. During the launch of the application, connection to the database is created. Right after, it is checked, either both of the database tables are already existing. If not, tables with respective attributes are then created. If the tables already exist, this step is skipped. Within the system, actions are performed over the data. After the successful execution of the algorithm, user can choose whether to save the data into the database or not. When selecting to save the data log into the database, username of a current logged in user is added as well. With the help of prepared statements and using inserts, each parameter is added to the respective database slot. It is also possible to access the data, using selects, which are necessary during most of the database methods. In other words, user can always open the database frame, which contains the history of saved sequence logs and information about them. Also, during the login process, the typed username and password are checked, if they match any database record. It is also checked whether the format of the inputs is correct. This also applies to the register process, which includes checking if password and confirm password content matches.

A logged in user can only access data logs, that were saved during the time, that he, specifically, was logged in, to make the data secured.

VI. REAL LIFE USE

The main use, which was addressed during the development of this application, was to create a helpful tool for the students studying the course on Data structures and algorithms. The main purpose is to better understand, practice and experience solving some tasks connected to the number sequences using various algorithms and to highlight the effectiveness of dynamic programming in the sphere of number sequences as well. Number sequences are very frequently used in the area of statistics. They are used in fields such biology, economics [23], geography, politics, signals [26], security [9], [27], [28] and others. Longest consecutive subsequences come in handy in all of these fields. For example, measurement of and estimation of the natural increase, price increase, number of deaths, average interest, fertility and many other important data can be accomplished, while important data can be collected. With some minor changes to the application, it would be easy to create a functionality, to find out, during what period of year, the average interest is at it's highest point, or what the longest period with the biggest natural increase was since the start of the records and so on.

One of the possible use of the application, is adjusting it the way, so that it can also process word strings, which would be essential to compare two thesis or diplomas, in order to find match and reveal plagiarism. An algorithm searching for the longest common subsequence of characters would be very effective in this area. It could also be used to find matches and interruptions between data logs, in order to find flaws in the system.

VII. CONCLUSION

The LCS problem is still very actual and a part of each data structures and algorithms courses. We gave an overview of how we teach these algorithms to students. In our teaching process, we use new and modern techniques that can help students to better understand some algorithms and concepts and make these topics more attractive. Our idea is to visualize some typical and interesting problems and to show particular steps while finding the solution to those problems. In this way, we prepared a software tool that provides a solving and visualization of finding the longest common subsequence of a given input. We want to continue in this work and to prepare some more complex software package for selected problems taught in course on Data structures and algorithms. We consider this will lead to increasing the attractivity and understandability of a course on Data structures and algorithms.

REFERENCES

- [1] R. van Aarle, and M. van der Giezen, 17 - Next-Generation Sequencing, Bioinformatics, and Infectious Diseases, Editor(s): Michel Tibayrenc, Genetics and Evolution of Infectious Diseases (Second Edition), Elsevier, ISBN 9780127999425, pp. 405–420, 2017, doi: 10.1016/B978-0-12-799942-5.00017-2.
- [2] A. V. Aho, D. S. Hirschberg, and J. D. Ullmann, Bounds on the complexity of the longest common subsequence problem. JACM, Vol. 23, pp. 112, 1976, doi: 10.1109/SWAT.1974.7
- [3] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, Data Structures and Algorithms, Addison-Wesley, 1983.

- [4] C. E. R. Alves, E. N. Cáceres, S. W. Song, A BSP/CGM Algorithm for the All-Substrings Longest Common Subsequence Problem. In Proceedings of the 17th International Symposium on Parallel and Distributed Processing (IPDPS '03). IEEE Computer Society, Washington, DC, USA, 57.1-., 2003, doi: 10.1109/IPDPS.2003.1213150
- [5] K. Baba, T. Nakatoh, T. Minami, Plagiarism detection using document similarity based on distributed representation, *Procedia Computer Science*, Vol. 111, pp. 382–387, 2017, doi: 10.1016/j.procs.2017.06.038.
- [6] K. Bringmann, and M. Künnemann, M.: Multivariate fine-grained complexity of longest common subsequence. In Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '18), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp.1216-1235, 2018.
- [7] K.-Y. Chen, S.-H. Liu, B. Chen, H.-M. Wang, H.-H. Chen, Exploring the use of unsupervised query modeling techniques for speech recognition and summarization, *Speech Communication*, Vol. 80, pp. 49–59, 2016, doi: 10.1016/j.specom.2016.03.006.
- [8] L. Chen, S. Goldwasser, K. Lyu, G. N. Rothblum, and A. Rubinfeld, Fine-grained complexity meets IP = PSPACE. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '19), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1–20, 2019.
- [9] D. Galinec, and W. Steingartner, A look at observe, orient, decide and act feedback loop, pattern-based strategy and network enabled capability for organizations adapting to change, In: *Acta Electrotechnica et Informatica*, Vol. 13, No. 2, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia pp. 39–49, 2013.
- [10] M. T. Hajiaghayi, S. Seddighin, and X. Sun, Massively parallel approximation algorithms for edit distance and longest common subsequence. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '19), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 1654–1672, 2019.
- [11] R. W. Hamming, Error detecting and error correcting codes, *The Bell System Technical Journal*, Vol. 29, No. 2, pp. 147–160, 1950. doi: 10.1002/j.1538-7305.1950.tb00463.x
- [12] D. S. Hirschberg, A linear space algorithm for computing maximal common subsequences, *Commun. ACM*, Vol. 18, No. 6 (June 1975), pp. 341–343, 1975. Doi:10.1145/360825.360861
- [13] D. S. Hirschberg, Algorithms for the Longest Common Subsequence Problem, *Journal of the ACM (JACM)*, Vol.24, No. 4, pp. 664–675, 1977.
- [14] P. Hogeweg, The Roots of Bioinformatics in Theoretical Biology, *PLoS Comput Biol* Vol. 7, No. 3, 2011, doi:10.1371/journal.pcbi.1002021
- [15] W.-J. Hsu, and M.-W. Du, New algorithms for the LCS problem, *Journal of Computer and System Sciences*, Vol. 29, No. 2, pp. 133–152, 1984.
- [16] W.-H. Hsu, Y.-Y. Chiang, Y.-S. Wu, Integrating weighted LCS and SVM for 3D handwriting recognition on handheld devices using accelerometers, *W. Trans. on Comp.*, Vol. 9, No. 3 pp. 235–251, 2010.
- [17] J. W. Hunt, and T. G. Szymanski, A fast algorithm for computing longest common subsequences. *Commun. ACM*, Vol. 20, No. 5, pp. 350–353, 1977. DOI:10.1145/359581.359603
- [18] M. Kellis, *Computational Biology: Genomes, Networks, Evolution* MIT course 6.047/6.878, 2016, Online: accessed 02-August-2019, url: https://ocw.mit.edu/ans7870/6/6.047/f15/MIT6_047F15_Compiled.pdf.
- [19] D. E. Knuth, J. H. Morris, and V. R. Pratt, Fast Pattern Matching Algorithms, *SIAM J. COMPUT.* Vol. 6, No. 2, 1977, pp. 323–350.
- [20] A. Koložvari, R. Stojanović, A. Zupan, E. Semenkina, V. Stanovov, D. Kofjač, A. Škraba, Speech-recognition cloud harvesting for improving the navigation of cyber-physical wheelchairs for disabled persons, *Microprocessors and Microsystems*, ISSN 0141-9331, Vol. 69, pp. 179–187, 2019, doi: 10.1016/j.micpro.2019.06.006
- [21] S. K. Kumar, and C. P. Rangan, A linear space algorithm for the LCS problem, *Acta Informatica*, Vol. 24, No. 3, pp.: 353–362, 1987, doi: <https://doi.org/10.1007/BF00265993>
- [22] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics - Doklady*, Vol. 10, No. 8, pp. 707–710, 1966.
- [23] E. Malichová, M. Ďurišová, and E. Tokarčíková, 4. Models of Application Economic Value Added in Automotive Company, *Transport Problems*, Vol. 12, No. 3, pp. 93–102, 2017.
- [24] C. D. Manning, P. Raghavan, and H. Schütze., *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2008.
- [25] D. Nath, J. Kurmi, and V. Rawat, A Survey on Longest Common Subsequence, *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Vol. 6, No. 4, pp. 4553–4557, 2018.
- [26] L. Ovseník, J. Turán, T. Vaniga, and P. Ivaniga, Deployment of the PON with an Optical Fibre G-652.B, 2017 IEEE 14th International Scientific Conference on Informatics, Nov 14-th-16th, Poprad, Slovakia, IEEE, pp. 289–292, 2017.
- [27] J. Perháč, and D. Mihályi, Coalgebraic modeling of IDS behavior, In: 2015 IEEE 13th International Scientific Conference on Informatics, Nov 18th-20th, Poprad, Slovakia, IEEE, pp. 201–205, 2015.
- [28] J. Perháč, D. Mihályi, and L. Mataš, Resource oriented BDI architecture for IDS, 2017 IEEE 14th International Scientific Conference on Informatics, Nov 14-th-16th, Poprad, Slovakia, IEEE, pp. 293–298, 2017.
- [29] M. Radovanović, A. Nanopoulos, M. Ivanović, Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, Vol. 11, pp. 2487–2531, 2010.
- [30] R. W. Soukoreff, and I. S. MacKenzie, Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*, ACM, New York, NY, USA, pp. 319–320, 2001, DOI: 10.1145/634067.634256.
- [31] W. Steingartner, V. Novitzká, M. Bačíková and Š. Korečko, New approach to categorical semantics for procedural languages, *Computing and Informatics*, Slovak Academy of Sciences, Bratislava, Slovakia, Vol. 36, No. 6, pp. 1385–1414, 2017.
- [32] F. S. Tabataba, and S. R. Mousavi, A hyper-heuristic for the Longest Common Subsequence problem, *Computational Biology and Chemistry*, ISSN 1476-9271, Vol. 36, pp. 42–54, 2012, doi:10.1016/j.compbiolchem.2011.12.004
- [33] R. A. Wagner, and M. J. Fischer, The String-to-String Correction Problem, *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.
- [34] Curricula Computer science, Faculty of Sciences, University of Novi Sad, Online: accessed 02-August-2019, url: <https://www.pmf.uns.ac.rs/studije/studijski-programi/racunarske-nauke-2017/>
- [35] Curricula Computer Science and Engineering, MIT, USA, Online: accessed 02-August-2019, url: <http://catalog.mit.edu/degree-charts/computer-science-engineering-course-6-3/>
- [36] Curricula Informatics, Faculty of Electrotechnical engineering and informatics, Technical University of Košice, Slovakia, Online: accessed 02-August-2019, url: <http://www.fe.tuke.sk/uploads/6c/62/6c62c822dbdd663babb312e58c05586e/informatics.pdf>
- [37] Bachelor's Program, Department of Computer Science, ETH Zurich, Switzerland, Online: accessed 02-August-2019, url: <https://inf.ethz.ch/studies/bachelor.html>
- [38] Database of protein domains, families and functional sites, Online: accessed 02-August-2019, url: <https://prosite.expasy.org/>
- [39] ENCODE: Encyclopedia of DNA Elements Online: accessed 02-August-2019, url: <https://www.encodeproject.org/>

Software visualization of natural semantics of imperative languages - a teaching tool

1st William Steingartner
*Faculty of Electrical Engineering
 and Informatics*
Technical University of Košice
 Košice, Slovak Republic
 william.steingartner@tuke.sk

2nd Martin Haratim
*Faculty of Electrical Engineering
 and Informatics*
Technical University of Košice
 Košice, Slovak Republic
 haratimmartin@gmail.com

3rd Jiří Dostál
*Department of Technical Education
 and Information Technology*
Palacký University Olomouc
 Olomouc, Czechia
 j.dostal@upol.cz

Abstract—In this work, we discuss the role and need of a teaching tool for the visualization of natural semantics method of imperative programming languages. The design, development and use of a compiler that can interpret simple programming language *Jane* statements to a tree that represents the semantic method of natural semantics is presented. A formal definition of *Jane* programming language and production rules in natural semantics is showed, as well. The developed teaching tool can provide particular visual steps in the process of finding the meaning of well-structured input program.

Index Terms—*Jane* language, lexer, natural semantics, parser, visualization

I. INTRODUCTION

The teaching of formal foundations of software engineering is nowadays a big challenge. It seems to be useful to show the future IT experts the advantages of formal methods and their fruitful usage in a process of practical software development [3], [13], [18]. One of the courses focused on formal methods in software engineering is a course on the semantics of programming languages. In this course, mostly the methods focused on imperative (and often also functional) languages are presented: operational semantics, denotational semantics and others.

With some techniques, the teaching of formal methods can be more attractive and more understandable for students [9]. For instance, formulating the modular structural operational semantics [11] where the descriptions of functional programs given in this method could be automatically translated into programs in the logic programming language Prolog, using the software provided to the students. Another approach was defining an action semantics [10] – a pure formal framework for describing the meaning of programs in textual phrases that are nearer to real programming languages and, moreover, quite good understandable for programmers that are not very familiar with mathematical methods.

The very fruitful method seems to be a visualization of processing the semantic method. We prepared more tools for visualization of some semantic methods, for instance, a

This work has been supported by the Slovak Research and Development Agency under the contract No. SK-AT-2017-0012: Semantics technologies for computer science education.

tool for categorical semantics [20] (categorical denotational semantics we formulated in [17], a tool for handling the mathematical expressions [19] or a tool for complex work with the abstract implementation of imperative language with an abstract machine for structural operational semantics [6].

The structure of a paper is the following: in Section II, we present some basic ideas and standard notions as a starting point. Furthermore, we show here the definition of a toy language *Jane* and we present how to define a natural semantics for this language. In Section III, the structure of a teaching software is presented: we briefly show a methodological part of the design, grammar for parsing the input source codes, and the role of particular phases of a program run. Section IV, an example of using the program is showed. The paper ends with a Conclusion section.

II. BASIC CONCEPTS

In this paper, we deal with a simple toy programming language *Jane*. Its syntax is mostly adopted from the well-known language *While* [12], or sometimes referred also as *IMP* [15]. The programming language needs a formal definition. Besides, it should be possible to mathematically justify programs, and this requires formal verification or proof of program behavior [8], [14].

A. Language, Syntax and Semantics

The basis of every language implementation is a language definition. The formal definition of a language consists of a

- syntax (structure) and
- semantics (meaning).

The syntax of a language determines which character strings constitute well-formed programs in the language and which do not: it is a “description” of how different parts of a language can be combined and other parts of that language created [8], [14]. The programming language syntax determines the design and structure of programs written in this language [2], [4], [14]. We define the syntax by

- with grammar;
- extended Backus-Naur form (BNF);
- inductive definition.

The semantics of a language describes the meaning of a program in terms of the basic concepts of a language [21]. The definition of formal semantics includes [14]:

- semantic domains,
- specification of semantic functions,
- semantic equality or deduction rules.

The semantic domain is a (mathematical) structure that contains the elements representing meanings of elements from a given syntactic domain [14]. To preserve transparency and ease, we will use only simple sets (\mathbb{Z} , \mathbb{N} , \mathbb{B} sets of integers, natural numbers and Boolean values) and sets that arise by applying operations to these sets [14]. The semantic function displays the syntactic area in the corresponding semantic area. We write its specification as [14]:

$$f : \text{Synt} \rightarrow \text{Sem.}$$

We assign just one semantic function to each syntactic domain. Semantic function is mostly given by:

- semantic equations (mainly for expressions), and
- production rules (for statements, see Section II-C).

We determine in the production rules the meaning of each form for a given syntactic area [14]. In this software tool, the most relevant for design of compiler is context-free grammar [1], [7] which we will use later in the implementation.

B. Formal definition of Jane programming language

In the previous section, we explained how a programming language is defined. In this section, we define the language that is used to teach the course on the semantics of programming languages. As other model languages for teaching, this language belongs to the imperative paradigm. Its syntax is adopted from the well-known toy language *While* [12] and we refer to this language as *Jane*.

A simple programming language *Jane* has the following syntax domains:

- **Num** – for numeric strings;
- **Var** – for variables;
- **Expr** – for arithmetical expressions;
- **Bexp** – for Boolean expressions;
- **Statm** – for statements.

The domains for numerals (**Num**) and variables (**Var**) have no internal structure from the semantic point of view. For the three remaining domains, particular production rules describing the syntax are defined.

For the arithmetic expressions, we formulate the following production rule:

$$e ::= n \mid x \mid e + e \mid e - e \mid e * e \mid (e), \quad (1)$$

where

- n denotes an integer numeral;
- x stands for a program variable;
- $e \bullet e$ represents an arithmetic operation that can be applied to arithmetic expressions (here: addition (+), subtraction (-), multiplication (*));

- (e) represents an arithmetic expression enclosed into parenthesis.

The Boolean expressions of the language *Jane* are given by the following production rule. In the case of language *Jane*, their role is to provide a logical condition in a conditional or a loop statement.

$$b ::= \text{false} \mid \text{true} \mid \neg b \mid b \wedge b \mid e = e \mid e \leq e \mid (b), \quad (2)$$

where

- **false**, **true** represent syntactic forms of Boolean constants;
- $e = e$ represents an equality of arithmetic expressions;
- $e \leq e$ represents a relation "less then or equal" of arithmetic expressions;
- $\neg b$ stands for a negation of a Boolean expression;
- $b \wedge b$ is a conjunction of Boolean expressions;
- (b) represents a Boolean expression enclosed into parenthesis.

The language *Jane* contains five (standard) imperative constructs [14]:

- a variable assignment statement;
- an "empty" statement;
- a pattern of sequencing the statements;
- a conditional statement with two mandatory ways of control-flow; and
- a loop statement.

The syntax of statements is given by the following production rule:

$$S ::= x := e \mid \text{skip} \mid S; S \mid \text{if } b \text{ then } S \text{ else } S \mid \text{while } b \text{ do } S. \quad (3)$$

These five commands of the imperative programming language are also referred to as Dijkstra commands (or D-diagrams) [14]. Based on this abstract syntax, we can define standard imperative constructs in real imperative languages. In the following sub-section, we present a natural semantics of the simple imperative language *Jane*.

C. Natural semantics production rules

We briefly express the basic properties of natural semantics of imperative languages, we list standard rules of this method and we show how this method is applied for the language *Jane*. We note that natural semantics is considered as a hybrid of operational and denotational semantics that shows computation steps performed in a compositional manner. It is also known as a "big-step semantics". This method has been proposed that is halfway between operational semantics and denotational semantics [16]. Like structural operational semantics, natural semantics shows the context in which a computation step occurs, and like denotational semantics, natural semantics emphasizes that the computation of a phrase is built from the computations of its sub-phrases.

So, the main role of natural semantics is to define the relationship between the initial state before executing the language statement and the final state after executing this

statement. The meaning of the statement (its semantics) is therefore considered as a change of state. Natural semantics does not follow the detailed execution of the statement; it is focused on changing the state that occurs by executing the entire statement, as it expresses the transition of natural semantics [14]:

$$\langle S, s \rangle \rightarrow s'.$$

The general form of the production rule of natural semantics looks as follows [14]:

$$\frac{\langle S_0, s_0 \rangle \rightarrow s_1, \dots, \langle S_n, s_n \rangle \rightarrow s}{\langle S, s_0 \rangle \rightarrow s} \quad (n_{ns})$$

where

- S stands for a statement, possibly consisting of sub-constructs S_1, \dots, S_n ;
- s_0 is an initial state;
- s is a final state;
- s_1, \dots, s_{n-1} are states during the particular steps;
- in the notation (n_{ns}) of a rule, n represents name or number of the rule and an index ns indicates, that it is a rule of natural semantics.

Notation of a general production rule (n_{ns}) expresses the natural semantics of the statement S as a change from the initial memory state s_0 to the final state s [14]. Now, we briefly introduce the rules of natural semantics for the statements of the language *Jane* listed in (3).

The assignment statement is defined by the axiom

$$\frac{}{\langle x := e, s \rangle \rightarrow s [x \mapsto \mathcal{E}[[e]]s]} \quad (1_{ns})$$

The rule (1_{ns}) above notation represents the meaning of the assignment command: the variable x is assigned to the value of an arithmetic expression e calculated in the state s and a memory state is being actualized from s to s' .

An empty command is defined by the axiom:

$$\frac{}{\langle \text{skip}, s \rangle \rightarrow s} \quad (2_{ns})$$

Here, the state is not being changed after the execution of a statement.

The sequence of commands is defined by the following rule:

$$\frac{\langle S_1, s \rangle \rightarrow s'' \quad \langle S_2, s'' \rangle \rightarrow s}{\langle S_1; S_2, s \rangle \rightarrow s'} \quad (3_{ns})$$

This rule expresses that commands are executed in particular steps with passing the actual memory state.

The conditional statement is defined by the following two rules:

$$\frac{\langle S_1, s \rangle \rightarrow s' \quad \mathcal{B}[[b]]s = \mathbf{tru}}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'} \quad (4_{\mathbf{tru}})$$

$$\frac{\langle S_2, s \rangle \rightarrow s' \quad \mathcal{B}[[b]]s = \mathbf{fls}}{\langle \text{if } b \text{ then } S_1 \text{ else } S_2, s \rangle \rightarrow s'} \quad (4_{\mathbf{fls}})$$

For the conditional statements, two rules based on the value of a Boolean conditions are defined. They are symmetric and their use is straightforward.

The loop statement is defined by the following two rule:

$$\frac{\langle S, s \rangle \rightarrow s'' \quad \langle \text{while } b \text{ do } S, s'' \rangle \rightarrow s' \quad \mathcal{B}[[b]]s = \mathbf{tru}}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'} \quad (5_{\mathbf{tru}})$$

$$\frac{\mathcal{B}[[b]]s = \mathbf{fls}}{\langle \text{while } b \text{ do } S, s \rangle \rightarrow s'} \quad (5_{\mathbf{fls}})$$

Similarly as in the case of a conditional statement, a value of a Boolean expression determines the way of execution of a loop.

Generally, a program in *Jane* is considered as a sequence of statements, i.e. one compound statement. For simplicity, we can denote the whole program by one statement-variable, e.g. P . Deriving the semantics of a program P we start in an initial state, e.g. s_0 and we construct a transition relation to a final state s . Such defined relation is a root of a derivation tree in natural semantics of a program P :

$$\langle P, s_0 \rangle \rightarrow s.$$

Then

- the transition of the program will form the root;
- by applying the rules of natural semantics, we create inner vertices;
- the leaves will form the axioms of natural semantics.

By step-wise applications of appropriate semantic rules, we construct a full derivation tree of a program in natural semantics.

In the next section, we present a software tool for constructing the derivation trees of natural semantics for programs prepared in a toy language *Jane*.

III. SOFTWARE TOOL IMPLEMENTATION

The program is designed to help students better understand how to interpret the meaning of a simple toy programming language code with the semantic method of natural semantics. The main idea is that the program cannot be difficult, it must be simple and transparent to use so that every student can work with it. Since the program reads an input source in *Jane*, it is designed and developed simply as a compiler transforming the source text into the semantic-driven visualization.

The first step in writing the compiler is a lexical analysis of the input string. This means dividing the input string into short units that represent the basic entities of a program. The lexical analysis is based on a class *Tokenizer* with an internal class *TokenInfo* which contains information about each token. The *TokenInfo* class contains two fields: an array containing a regular expression that is intended to find a match between an input string and a token that is stored in the *pattern regex* field. Instead of a regular expression, we store sample objects to help improve performance. We define a linked list *tokenInfos* that stores information about all tokens. The linked list is created in the constructor of our class *Tokenizer*. Next, we need to

add token information to our list. We realize this step by using our method *add()*. The user can send arguments to this method with a regular expression string and a token code. The method then adds “^” to the user-supplied regular expression. This causes the regular expression to look only for the beginning of the string. This is necessary because we successively remove each token looking for another token at the beginning of the input string. We also want to store the tokens we’ve already scanned. We need to save the code and string that matches this token. Moreover, we need to keep a string because the token code does not retain complete input information. When we find a variable at the input, we give it a special code for the token variable, but we also need to store its name so that we can use it later to set or retrieve the value. To do so, we define another internal class *Token* and another linked list of these tokens. We create a method *tokenize()*. In this method, we define first the local variable which is supposed to contain the input string but without any unnecessary spaces. We also clear content of lists from any previous data. During the method execution, we successively remove the parts of the chain that match the pattern. Then string must be empty when the method is over. If everything went fine, we have a list of tokens for further processing at the end. The next step is to create a grammar to analyze commands and expressions. Grammar is a set of rules that specify how input is structured [2]. Before we begin to explain the rules, we have to say what we apply for the rules. The tokenizer produces a token sequence and assigns a unique code to each token type. These token codes are considered as grammar terminal symbols. In addition to terminal symbols, we also define non-terminal symbols.

The class *Parser* serves for analyzing the expressions. Tokens are stored in the *Token* list and one token from that list is stored as *lookahead*. In the main method *Parse*, we first create a copy of the tokens because we select elements from the list and do not want any side effects on the original list. The first token in the list is then assigned to the *lookahead*. After these initial functions, the *commands()* method is called. This method represents the first non-terminal grammar symbol, that is, the start symbol.

The parser handles one method for each non-terminal grammar symbol created in the previous paragraph. Once the parsing process is completed in the list, there should be no more symbols. Finally, parsing should reach non-terminal *EPSILON*. If there is still a token in the *lookahead*, it means there is something unexpected at the input. After the program is parsed, we can check for errors. With this check, we ensure the correct number of parentheses, semicolons, and similar symbols that make up the syntactically and semantically correct program. We use the *nextToken* method to reduce the found terminal symbols. The functionality is elementary, remove the first token from the list, and set a new first token that follows in the list. If the list is empty, insert a non-terminal symbol *EPSILON*, which signals the parser that the input terminated.

The final step in implementation is to design a graphical interface. The interface was implemented using external *JavaFX*

and *JLTeXMath* libraries. Using the first one, components were created to interact with the user, and the second allowed to render a graphical representation of the specified source code on the tree of natural semantics.

Our grammar is defined as follows:

```

commands → sub_commands ; { sub_commands ; }
sub_commands → assign | statement | cycle
sub_commands → EPSILON
assign → VARIABLE ASSIGNMENT expr
assign → EPSILON
statement → IF (expr) then commands
               { else commands }
statement → EPSILON
cycle → WHILE (expr) do commands
cycle → EPSILON
expr → sig_operator log
log → PM operator log
log → EPSILON
sig_operator → PM operator
sig_operator → operator
operator → sig_term sum
sum → PM term sum
sum → EPSILON
sig_term → PM term
sig_term → term
term → factor product
product → MD sig_factor product
product → EPSILON
sig_factor → PM factor
sig_factor → factor
factor → value
factor → NEG factor
factor → O_BR expr C_BR
value → NUM
value → VAR

```

The resulting GUI look can be seen in Fig. 1.

IV. THE USE OF APPLICATION

In this section, we briefly present how this software is used. When launched, the main application window appears, which contains the components to interact with. At the top of the window, we can see the menu bar where you can work with all the features app has to offers. Right below this panel, there is a text area into which input code can be written. Below this text area is a set of buttons that perform the following tasks:

- *Generate* button – it calls the main program method and generates application output.
- *Save Input* button – it is used to store input to a user-selected location in the computer’s memory.
- *Set Variables* button – it allows user to declare variables before the output is generated.
- *Clear Input* button – it is to clear the text area.

After entering the input string in the form of a *Jane* code and possibly setting the variables and pressing the *Generate* button, a new component appears at the bottom of the screen.

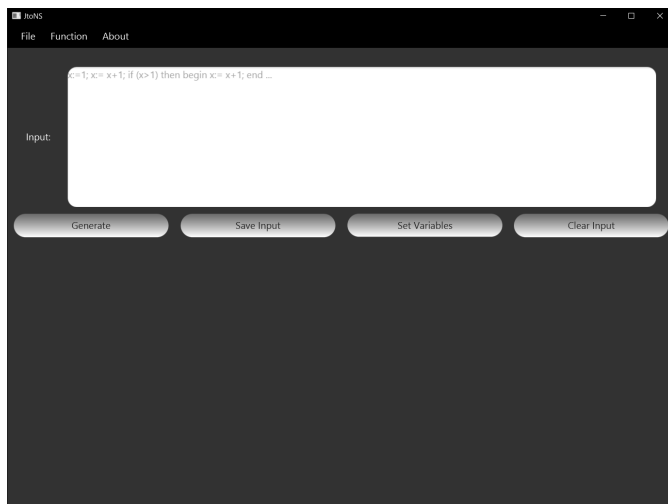


Fig. 1. Graphical User Interface of teaching tool

This component contains a canvas to render the resulting

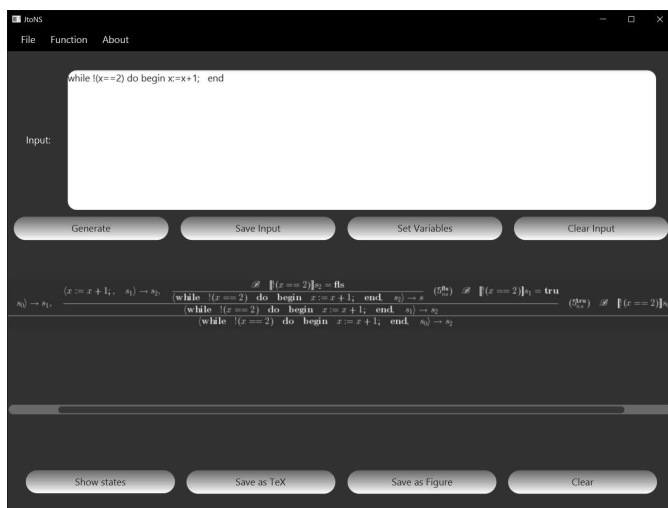


Fig. 2. Application after output generation

natural semantic tree and set of buttons. The set consists of the following buttons:

- *Show States* button – it displays a pop-up window in which the table shows the states that the program has passed during execution.
- *Save as TeX* button – it saves the latex-code output string to a text file on a user-selected location in the computer’s memory.
- *Save as Figure* – similar to the previous button, it saves the result to the computer’s memory but in the image format of the user’s choice. Supported formats are *JPG* or *PNG*.
- *Clear* – it removes the entire component that was created after pressing the *Generate* button.

Accessing these functions is also possible from the upper menu bar.

As an example of using this program, we show how our software tool produces a visual output for the Euclidean algorithm finding the greatest common divisor of two given numbers.

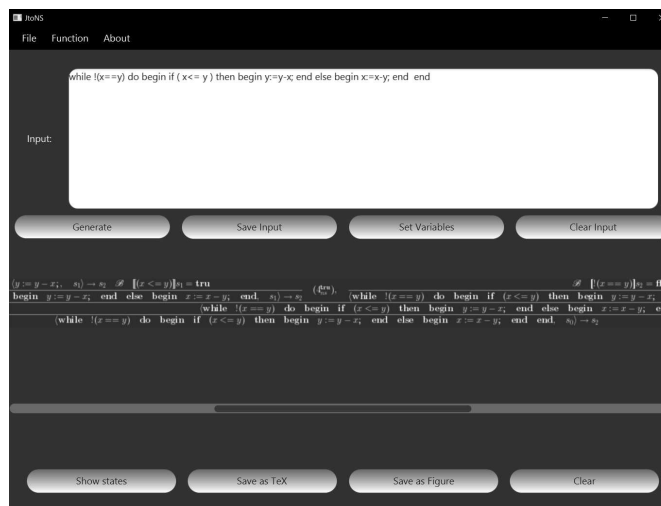


Fig. 3. Example of using the tools with implementation of Euclidean algorithm

In the text area, we insert the program realizing the Euclidean algorithm in the form of *Jane* code. We wrote the program into the text area, then we set the initial values for variables *x* and *y* to **96** and **64**, resp.

A program given as an input is the following:

```
while !(x == y) do begin
  if (x <= y) then begin y := y - x end
  else begin x := x - y end
end
```

We note, that for easier manipulation with the source and better readability:

- we enriched the blocks in a loop statement and a conditional statement with the keywords (textual “brackets”) **begin** and **end**; and
- we allowed to use C-like syntax for Boolean expressions (in contrast to rule (2)): == for equality of expressions, <= for less-or-equal symbol, ! for negation and && for logical conjunction.

The result can be seen in Fig. 3. Moreover, we can see the states that occurred during the program functioning in the table. When user recalls the states (using the *Show States* button), the table appears with all the states (Fig. 4).

V. CONCLUSION

In this paper, we presented a software tool that allows visualizing the construction of trees in natural semantics of imperative programming languages. The method of natural semantics is implemented on a toy language *Jane* using for

State number	Variable name	Variable value
State 0	x	96
State 0	y	64
State 1	x	32
State 2	y	32

Fig. 4. States of Euclidean algorithm

teaching the principles of languages and the semantics. Our tool fully supports all standard imperative constructs - variable assignment, sequencing of statements, conditional and loop statement. As an added value, the program allows the user to store the state table into a text file, to store a source code of an input program after modifications, to store the picture of a depicted tree and to produce a source code of a drawn tree for \LaTeX . A presented software tool is planned to be integrated into a larger software package for teaching the semantic methods. The main motivation for creating this tool was mainly help for students and technically interested persons and young IT experts to get more familiar with formal methods grounded in the semantics of programming languages. Our main goal was to visualize the semantic method and allow to understand quickly (and easily) read its results.

REFERENCES

- [1] A. V. Aho, S. Ravi, and J. D. Ulman, *Compilers, principles, techniques, and tools*, Addison-Wesley Publishing Company, 1987.
- [2] L. Dederá, *Computer languages and their processing*, Liptovský Mikuláš, Slovakia, 2014.
- [3] J. Dostál, X. Wang, P. Naungchalerm, A. Brosch and W. Steingartner, Researching computing teachers' attitudes towards changes in the curriculum content — an innovative approach or resistance? In: ICIC 2017, New York, IEEE, pp. 1–6, 2017.
- [4] M. Gabrielli, S. Martini, *Programming languages: principles and paradigms*, Springer-Verlag London, 2010.
- [5] G. Kahn, Natural semantics, In: Brandenburg F.J., Vidal-Naquet G., Wirsing M. (eds) STACS 87. STACS 1987. Lecture Notes in Computer Science, vol 247. Springer, Berlin, Heidelberg.
- [6] Ž. Kochaníková, W. Steingartner, M. A. M. Eldojali, A code generator for an abstract implementation of imperative language, In: Electrical Engineering and Informatics 8 : proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, pp. 342-347, 2017.
- [7] J. Kollár, *Compilers*, elfa s.r.o., Košice, Slovakia, 2010 (in Slovak).
- [8] K. Louden, K. Lambert, *Programming languages - Principles and Practice*, Third edition, Cengage Learning, USA, 2011.
- [9] D. Mihályi, M. Peniašková, J. Perhác and J. Mihelič, WEB-Based Questionnaires For Type Theory Course, Acta Electrotechnica et Informatica, Technical University of Košice, Košice, Slovakia, Vol. 17, No. 4, pp. 35-42, 2017.
- [10] P. D. Mosses, *Theory and Practice of Action Semantics*, Aarhus, DK: University of Aarhus, 1996. BRICS Report Series RS9653.
- [11] P. D. Mosses, Teaching Semantics of Programming Languages with Modular SOS, In: Proceedings of the 2006 Conference on Teaching Formal Methods: Practice and Experience, Series TFM'2006, BCS Learning & Development Ltd., Swindon, UK, 2006.
- [12] H. Riis Nielson, F. Nielson, *Semantics with Applications: An Appetizer (Undergraduate Topics in Computer Science) 2007th Edition*, Springer, 2007.
- [13] V. Novitzká, Logical Reasoning about Programming of Mathematical Machines, Acta Electrotechnica et Informatica, Technical University of Košice, Košice, Slovakia, Vol. 5, No. 3, pp. 50-55, 2005.
- [14] V. Novitzká, W. Steingartner, Semantics of programming languages, Technical University of Košice, Košice, Slovakia, 2015 (in Slovak).
- [15] G. Roşu, K - A Semantic Framework for Programming Languages and Formal Analysis Tools, In: D. Peled and A. Pretschner (eds.), Dependable Software Systems Engineering, Series NATO Science for Peace and Security, IOS Press, 2017.
- [16] A. D. Schmidt, Programming language semantics, In: Encyclopedia of Computer Science, John Wiley and Sons Ltd., Chichester, UK, pp. 1463–1466, 2003.
- [17] W. Steingartner, V. Novitzká, M. Bačíková and Š. korečko, New approach to categorical semantics for procedural languages, Computing and Informatics, Slovak Academy of Sciences, Bratislava, Slovakia, Vol. 36, No. 6, pp. 1385-1414, 2017.
- [18] W. Steingartner, D. Radaković, V. Novitzká, and M. A. M. Eldojali, An analysis of some aspects of component-based programming for selecting appropriate categorical structures as their models, Acta Electrotechnica et Informatica, Technical University of Košice, Košice, Slovakia, Vol. 17, No. 2, pp. 3-10, 2017.
- [19] W. Steingartner, I. Yar-Muhamedov, Learning software for handling the mathematical expressions, Journal of Applied Mathematics and Computational Mechanics, Vol. 17, No. 2, Czestochowa University of Technology, Czestochowa, Poland, pp. 77-91, 2018.
- [20] W. Steingartner, J. Perhác and A. Biliński, A Visualizing Tool for Graduate Course: Semantics of Programming Languages, IPSI BgD Transactions on Internet Research, Belgrade, Serbia, Vol. 15, No. 2, pp. 52–58, 2019.
- [21] W. M. Waite, G. Goos, *Compiler Construction, Series: Texts and Monographs in Computer Science*, Springer-Verlag, 1984, reprint 1996.

Designing Voice-Controllable APIs

Matúš Sulír, Jaroslav Porubán
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
 Košice, Slovakia
 {matus.sulir, jaroslav.poruban}@tuke.sk

Abstract—The main purpose of a voice command system is to process a sentence in natural language and perform the corresponding action. Although there exist many approaches to map sentences to API (application programming interface) calls, this mapping is usually performed after the API is already implemented, possibly by other programmers. In this paper, we describe how the API developer can use patterns to map sentences to API calls by utilizing the similarities between names and types in the sentences and the API. In the cases when the mapping is not straightforward, we suggest the usage of suitable annotations (attribute-oriented programming).

Index Terms—application programming interface (API), methods, parameters, voice control, natural language commands

I. INTRODUCTION

Voice control is slowly becoming a mainstream form of human-computer interaction. For this reason, developers often integrate voice control into the applications being developed. In this article, we will focus on simple, one-sentence commands after which we expect the computer to perform the desired action. Such a process consists of two main parts [1]: the translation of the voice input to a natural language sentence and the selection of an appropriate action according to the meaning of the sentence. In this paper, we suppose the voice recognition is implemented by a third-party library or service. Therefore, we will focus on the second part: the mapping of a sentence to the desired action.

Since we are interested in the implementation of the voice control from the developer's view, by an action we mean a call to an appropriate method in an API (application programming interface) with correct parameters. There already exist many approaches to map a natural language sentence to API calls. Some of them utilize probabilistic grammars and heuristics [2], others perceive phrases only as lists of keywords [3]. An increasingly popular approach is the application of machine learning to translate natural language sentences to API snippets [4].

The common sign of these approaches is that they perceive voice control only as an afterthought – when the API is already implemented, probably by other developers. In contrast to them, we will show how the API can be tailored to be voice-controllable, predominantly by exploiting the similarity of names and data types used in the sentences and the API. For example, a sentence “open the door for 5 seconds” could be interpreted as a call to `door.open(5)` during the execution of a program, provided there is a method `void open(int`

seconds) in the class `Door`. When such similarities are not observed, we suggest utilizing annotations (attribute-oriented programming) to specify the necessary details of the sentence-to-API mapping.

Throughout the paper, we also discuss how such APIs could be analyzed by a voice command recognition framework, which would then execute an appropriate action given a natural language sentence. A prototype implementation of such a framework, supporting many of the described features, is available online¹.

II. MAPPING PATTERNS

In this section, we will gradually describe individual patterns of mapping between a natural language sentence and an API call. Our examples will be related mainly to the hardware device control domain, but it is not restricted to it. Although we will use Java, the approach is generally applicable to any object-oriented programming language with sufficient metaprogramming features, static typing is also helpful for some patterns.

A. Verbatim Name Mapping

Suppose we would like to turn on light in a room after saying “turn on light”. The most primitive way to map this voice command to a method in the API would be to name the method `turnOnLight` and annotate it with a marker annotation `VoiceControllable`, so our framework could recognize it as a voice-controllable function:

```
@VoiceControllable
public static void turnOnLight() {
    // implementation turning on the light...
}
```

The name of the method would be split into individual words and transformed to lowercase. Using exact string matching, the input “turn on light” could be then mapped to this method, which would be executed without any parameters.

Of course, having one class with many static methods is a bad programming practice. Therefore, suppose we have separate classes to control lights, monitors, speakers, and other devices. For simplicity, let us say these classes are named `LightService`, `MonitorService`, etc. In modern application frameworks, service constructors are usually not called

¹<https://github.com/sulir/voice-control-demo>

manually – the framework initializes them using techniques such as dependency injection. Thus we will assume we have an instance of every necessary class readily available and we will focus on calling methods of this object. We can transform the previous example to this code:

```
@VoiceControllable
public class LightService {
    public void turnOn() {
        // implementation turning on the light...
    }
}
```

The annotation over a class means all its methods are now “voice-controllable”. Our framework searches for all such classes and their methods, building potential command-to-method mappings. Since `Service` is an implementation-oriented name irrelevant to the problem domain [5], our framework strips it from the class name when scanning it. The rest of the class name is appended to the method names. Therefore, the `turnOn` method of class `LightService` will be called when the command “turn on light” is issued.

B. Minor Variations

Since natural language is not restricted to exact phrases, we must count with some variations in the commands. First, we should stem all words in both the command and the class/method names using an appropriate stemmer. This way, we allow for variation in individual words since only their root forms are compared. For example, the command “turn on lights” can now execute the aforementioned method, even if “Light” is singular in the class name.

Next, we should allow permutations of the words. For example, “turn on lights” can be expressed also as “turn lights on”. In general, we need to find a compromise between allowing all permutations (which allows diverse sentences at the price of higher ambiguity) and supporting only certain kinds of permutations.

We can also allow extraneous words to be present in the command. Typical examples include stop words (“the” in “turn on the light”) or courtesy phrases (“please turn on the light”). An extreme example is the sentence “Please be so kind and turn on the light, dear.” As long as the extraneous words are not contained in the names in our API, we can safely remove them from the sentence without any ambiguity. To enable more precise decisions, we can annotate the method with words we expect to encounter in the command. For example, we would like to execute the method `turnOff` by the command “turn off all lights”:

```
@ExtraWord("all")
public void turnOff() { ... }
```

A similar case is the non-presence of certain words from API names in the natural language command. For instance, the command “light off” does not contain the word “turn” from the method `LightService::turnOff`. To help the mapping system to make a correct decision, we can annotate the method with `@OptionalWord("turn")`.

By allowing all of the mentioned variations at once, we can achieve a system which is highly flexible and accepts a wide range of different sentences – at the price of potential ambiguity. Essentially, the framework should find a method in the API which has the highest number of words common with the input. When there are ties or the number of common words is too low, we can start taking the `ExtraWord` and `OptionalWord` annotations into account. If the situation is still indecisive, the voice assistant can simply list the most probable options and ask the person to select one of them.

C. Parameters

Now we will take into account methods with parameters. The simplest case is a method with one numeric parameter:

```
public void turnOn(int number) { ... }
```

Typical commands to execute this method are: “turn on light number 1” and “turn on light 1”. In the first sentence, the argument value follows the parameter name “number”, which means the argument can be matched by its name. In the second case, the parameter is matched only by its type – the input sentence contains a number and the API method has a numerical parameter.

Mapping enum-typed parameters is straightforward too. Suppose `Position` is defined as an enumeration:

```
public enum Position {
    LEFT, MIDDLE, RIGHT
}
```

We also have a method with a `Position` parameter:

```
public void turnOn(Position position) { ... }
```

Then this method can be called by the command “turn on the left light”. On the other hand, mapping of string parameters is more problematic:

```
public void turnOn(String name) { ... }
```

If the user says “turn on the light named ‘front’”, the system can use the matching based on parameter names. However, mapping the sentence “turn on the front light” to this method is questionable in general, since we do not have any way to enumerate all possible valid values of a `String` parameter. A potentially dangerous way to determine if this method matches the command is trying to execute it and observing if it does not throw an exception. There exist better ideas, though. If the list of valid values is fixed at a compile time, it can be supplied directly:

```
public void turnOn(
    @Values({"front", "back"}) String name
) { ... }
```

In such cases, however, it is much better to use enumerations instead. Most often, the set of valid values can be enumerated only at runtime. Then we can create a class with a method returning these values:

```
public class LightNames implements ValueSet<String> {
    public Set<String> getValues() {
        return configuration.getValidLightNames(...);
    }
}
```

```
    }
}
```

Subsequently, we connect this list with a parameter using an annotation:

```
public void turnOn(
    @ValidValues(LightNames.class) String name
) { ... }
```

Now we will consider parameters of any class in general, e.g., a parameter of type `Color`:

```
public void setColor(Color color) { ... }
```

In the command “set light color to green”, we do not know how to map “green” to `Color(0, 255, 0)`. It is possible that `Color` has a constructor accepting a string, in which case we can utilize it. However, this still does not solve the problem of the enumeration of all valid values described in the previous example. Therefore, we suggest using a string-to-object mapping via annotations. Each parameter of a custom class should be annotated with a mapper:

```
public void setColor(
    @StringMapping(ColorMapper.class) Color color
) { ... }
```

Then the `ColorMapper` class can look like this:

```
public class ColorMapper
implements StringMapper<Color> {
    public Map<String, Color> getMap() {
        return Map.of(
            "red", new Color(255, 255, 0),
            "green", new Color(0, 255, 0),
            ...
        );
    }
}
```

This way, we can both enumerate all possible values of a `Color`-typed parameter and map a string such as “green” to a `Color` object. The approach is not limited to a constant associative array of string–color pairs: the mapping could be also dynamically generated by arbitrarily complicated code. Note that since the mapping is specified in a separate class, we did not modify the original `Color` class in any way – it can be defined in a third-party library without problems.

Next, we consider a method with multiple parameters:

```
public void setColor(int number, Color color) { ... }
```

As long as the parameters are of distinct types and the sets of valid values of all types are disjunctive, we can map various commands to a method execution without too much ambiguity: e.g., “set light 3 to blue”, “I would like yellow color for light 4”. However, methods with multiple parameters of same or similar type are more problematic:

```
public void setBrightness(int light, double brightness)
{ ... }
```

Here, “set light 1 to brightness 50” can be mapped to a correct execution using the parameter names. However, the sentence “set brightness of light number 2 to 30” could be probably

mapped only by the position of arguments, which would not work correctly if the parameters were switched in the API.

Note that parameters have an important role in the process of the API method selection. Since it is not valid to call a method without all mandatory (non-null) arguments filled, they must be all present in the input sentence. For example, if the command does not contain any number, we can assume it does not match any method with a numeric parameter.

D. Collections

Suppose we want to dim multiple lights at once. A collection, such as a set, is an ideal candidate for this:

```
public void dim(Set<Integer> lights) { ... }
```

The content of the parameter `lights` can be expressed by enumerating all values or by specifying a range: “dim lights 1, 7 and 9” or “dim lights 6 to 10”. In case the expected range can be very large, we recommend using a lazy collection denoted by an interface, such as `Iterable<Integer>`.

Similarly to numeric collections, we accept collections of enumerations or other classes:

```
public void dim(Set<Position> lights) { ... }
```

Naturally, ranges are not supported in such cases. We can still fill this set with values by naming them all, though: “dim the left and middle light”. Alternatively, commands such as “dim all lights” should work if there is a way to enumerate all valid values of the parameter programmatically.

E. Synonyms

Many words have synonyms which can a user utter instead of the words used in the API names. A natural way to cope with such situations is the usage of thesauri and lexical databases such as WordNet [6]. Any word in the API can be then replaced by its synonym to perform a successful match with the natural language command. However, consider the following example of a method in class `ScreenService` (controlling a multi-monitor ambient user interface [7]), where `State` is an enumeration with values `ON` and `OFF`:

```
public void set(int screen, State state) { ... }
```

We would certainly like to say not only “set screen 1 to ‘on’” but also “turn screen 1 on”. However, “turn” is not a synonym of “set” in general – only in this specific case. Therefore, we devised a way to specify method-local synonyms via annotations:

```
@Synonym(of = "set", is = "turn")
```

```
public void set(int screen, State state) { ... }
```

This synonym can be applied only during the matching of this particular method. Analogously, we support package-local, class-local and parameter-local synonyms using annotations over packages (in a special file `package-info.java`), classes, and parameters, respectively. It is also allowed to specify multiple synonyms per element, e.g.:

```
@Synonym(of = "screen", is = {"display", "monitor"})
```

```
@Synonym(of = "turn", is = "switch")
```

```
public class Screen { ... }
```

F. Fallback

If we encounter difficulties during the application of the aforementioned mapping patterns, we can always specify the voice commands manually. For example, suppose we have a class `SpeechService` with a method `pronounce(String sentence)` which says out loud the given sentence. Since the sentence can be completely arbitrary, matching will likely fail. Therefore, we specify the voice command as a regular expression:

```
@VoiceCommand("say (.*)")
public void pronounce(String sentence) { ... }
```

The content of the group in the parentheses will be supplied as a parameter value. Because a manually specified command has a high priority, we can successfully match commands such as “say ‘I like turning off the screens’” even when they contain words present in the API.

III. MAPPING PROCESS

In our prototype implementation², we decided to use a simple sentence recognition algorithm designed to allow for relatively large deviations of the input sentences from the prescribed forms. Now we will briefly describe it.

First, the input sentence is matched against all fallback regular expressions (section II-F). If a match is found, the process is stopped and the annotated method is executed.

Next, for each voice-controllable method, we try to type-match all its parameters with the sentence. For example, if a method has a numeric parameter and a `Color` enumeration parameter, the sentence is searched for a numeral and a word denoting a color. The result is a list of potentially matching methods, along with word-to-parameter mappings.

For each method in this list, a score of similarity with the input sentence is calculated: Let W_M be the set of words contained in the class and method name. Let W_S be the set of words contained in the input sentence, excluding the parameter values matched in the previous steps. The score is computed as the Jaccard index [8] of these two sets:

$$\frac{|W_M \cap W_S|}{|W_M \cup W_S|}$$

If the class or method is annotated by synonyms, multiple variants of the set W_M are constructed, with the words in the identifiers gradually replaced by their corresponding synonyms. The resulting score is a maximum of the scores computed for individual variants.

Finally, the method with the highest score is executed. If multiple methods have the same score or if no method receives a score higher than a certain threshold (e.g., 0.2), the user should be asked to reformulate the command.

It is important to note that the described algorithm is not the only one possible. The ideas described in the previous section are compatible, for instance, with strict regex-based solutions based on the precise matching of sentences with regular expressions generated from the identifier names and

annotations. Another possibility is a grammar-based approach, which would distinguish parts of sentences: Some of them could be derived automatically if possible, the rest will be determined by manually written annotations.

IV. RELATED WORK

Many approaches mapping sentences to API calls are grouped under the umbrella term of program synthesis, particularly program synthesis using natural language input [9]. Desai et al. [10] synthesize source code written in various domain-specific languages, thanks to a dataset of sentence–code pairs used for training. Gvero and Kuncak [2] synthesize Java expressions using probabilistic grammars and heuristics. In T2API, Nguyen et al. [4] perceive program synthesis as a statistical translation process from natural language to a programming language. Little and Miller [11] generate Java code from a set of brief keywords. Some synthesis approaches are focused on particular domains or technologies: e.g., SQL query generation [12], smartphone automation script synthesis [13], bot API invocations [14]. All of the mentioned approaches perceive APIs as black boxes, which are already designed. In contrast to them, our idea is to engage the API designers in the process of natural language command specification.

Landhäuser et al. [15] designed NLCI (natural language command interpreter), which has a goal similar to ours – to perform an action in the API, given a natural language sentence as an input. In contrast to us, they first transform the API into an ontology by analyzing the relationships between elements in the code and combining it with a general-purpose ontology. Furthermore, they do not support simple mapping customization via annotations.

The command execution approach by Little and Miller [3] is based on the similarity of names used in the sentence and the API. They also perceive sentences as lists of keywords, allowing for variations such as extraneous words. However, they do not allow any customization using annotations, since they consider APIs to be developed by a third party and thus not modifiable.

Naturalistic programming [16] is a paradigm aiming to make the source code look more like natural language. For example, Knöll et al. [17] discuss naturalistic types which include the mapping of natural language quantities such as “nearly all” to exact numeric intervals. Compared to them, we aim to integrate voice control with existing, traditional programming languages instead of designing new ones.

There exist guidelines on how to design APIs in general [18] and an overview of design decisions to be made when creating an API [19]. Nevertheless, none of these works take voice-controllability into account.

Hirzel et al. [20] describe an idea of grammars for dialog systems, including virtual voice assistants. However, they do not try to solve the problem of the mapping of sentences to API calls.

²<https://github.com/sulir/voice-control-demo>

Commercial system APIs, such as Google Voice Actions for Android³ or SiriKit⁴ are often limited to certain domains and action types. Furthermore, they require some effort to integrate, such as the creation of configuration files or implementation of non-trivial interfaces.

YAJCo [21] is a parser generator utilizing the similarity between the relations of program elements in Java source files and production rules of computer language grammars. The core ideas behind this article stemmed from YAJCo, however, this time they are applied to natural languages.

V. CONCLUSION AND FUTURE WORK

In this paper, we described patterns of mapping between a natural language command and an API method call. These patterns are based on:

- class, method, and parameter names,
- parameter types and positions.

The mapping can be enabled simply by placing the annotation `@VoiceControllable` over a class or a method. When these natural mapping patterns are not sufficient, the programmer can adjust the mapping by using annotations, such as:

- `@StringMapping(Mapper.class)` to specify the string-to-object mapping for a parameter,
- `@ValidValues(ValueSet.class)` to enumerate possible values of a parameter at runtime,
- `@Synonym(of="word1", is="word2")` over packages, classes, methods, and parameters to specify local synonyms,
- `@VoiceCommand("regex (param)")` to specify an exact regular expression whose match will execute the given method.

There are many limitations of the described work. First of all, we did not yet perform full validation of our ideas. We should create a golden standard of sentence-to-API mappings (or utilize an existing one). Then we need to validate our approach by measuring the accuracy of the algorithms based on our ideas when compared to the golden standard. We hypothesize our simple approach based on word similarities would work well for small or medium-sized APIs, but it could be problematic for larger ones.

Next, we described only a small portion of all useful patterns. In the future, we could devise more elaborate ways to express numerical ranges, binary operators, various collection types, exceptions, etc. String-to-object mappers could be improved too. In addition to the manual definition of synonyms, domain ontologies could be used too.

The examples mentioned in this article are very simple – in order to show the point of our approach. We should inspect larger APIs from multiple domains and assess the applicability of our patterns to them.

Finally, in this article, we were interested only in simple, one-sentence inputs being mapped to single API calls. The approach could be extended to support nested API calls or more complex natural language dialogs in the future.

³<https://developers.google.com/voice-actions/custom-actions>

⁴<https://developer.apple.com/documentation/sirikit>

ACKNOWLEDGMENT

This work was supported by Project VEGA No. 1/0762/19 Interactive pattern-driven language development. This work was also supported by FEI TUKE Grant no. FEI-2018-57 “Representation of object states in a program facilitating its comprehension”.

REFERENCES

- [1] A. Rogowski, “Industrially oriented voice control system,” *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 3, pp. 303–315, Jun. 2012.
- [2] T. Gvero and V. Kuncak, “Synthesizing Java expressions from free-form queries,” in *Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA 2015. ACM, 2015, pp. 416–432.
- [3] G. Little and R. C. Miller, “Translating keyword commands into executable code,” in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’06. ACM, 2006, pp. 135–144.
- [4] T. Nguyen, P. C. Rigby, A. T. Nguyen, M. Karanfil, and T. N. Nguyen, “T2API: Synthesizing API code usage templates from english texts with statistical translation,” in *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE 2016. ACM, 2016, pp. 1013–1017.
- [5] J. Kollár, E. Pietriková, and S. Chodarev, “Abstraction in programming languages according to domain-specific patterns,” *Acta Electrotechnica et Informatica*, vol. 12, no. 2, pp. 9–15, 2012.
- [6] G. A. Miller, “WordNet: A lexical database for English,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.
- [7] L. Galko and J. Porubán, “Tools used in ambient user interfaces,” *Acta Electrotechnica et Informatica*, vol. 16, no. 3, pp. 32–40, 2016.
- [8] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.
- [9] S. Gulwani, “Dimensions in program synthesis,” in *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming*, ser. PPDP ’10. ACM, 2010, pp. 13–24.
- [10] A. Desai, S. Gulwani, V. Hingorani, N. Jain, A. Karkare, M. Marron, S. R, and S. Roy, “Program synthesis using natural language,” in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE ’16. ACM, 2016, pp. 345–356.
- [11] G. Little and R. C. Miller, “Keyword programming in Java,” in *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE ’07. ACM, 2007, pp. 84–93.
- [12] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig, “SQLizer: Query synthesis from natural language,” *Proc. ACM Program. Lang.*, vol. 1, no. OOPSLA, pp. 63:1–63:26, Oct. 2017.
- [13] V. Le, S. Gulwani, and Z. Su, “SmartSynth: Synthesizing smartphone automation scripts from natural language,” in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’13. ACM, 2013, pp. 193–206.
- [14] S. Zamanirad, B. Benatallah, M. Chai Barukh, F. Casati, and C. Rodriguez, “Programming bots by synthesizing natural language expressions into API invocations,” in *Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2017. IEEE Press, 2017, pp. 832–837.
- [15] M. Landhäuser, S. Weigelt, and W. F. Tichy, “NLCl: A natural language command interpreter,” *Automated Software Engg.*, vol. 24, no. 4, pp. 839–861, Dec. 2017.
- [16] O. Pulido-Prieto and U. Juárez-Martínez, “A survey of naturalistic programming technologies,” *ACM Comput. Surv.*, vol. 50, no. 5, pp. 70:1–70:35, Sep. 2017.
- [17] R. Knöll, V. Gasiunas, and M. Mezini, “Naturalistic types,” in *Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, ser. Onward! 2011. ACM, 2011, pp. 33–48.
- [18] J. Bloch, “How to design a good API and why it matters,” in *Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications*, ser. OOPSLA ’06. ACM, 2006, pp. 506–507.

- [19] J. Stylos and B. Myers, “Mapping the space of API design decisions,” in *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*, ser. VLHCC '07. IEEE Computer Society, 2007, pp. 50–60.
- [20] M. Hirzel, L. Mandel, A. Shinnar, J. Simeon, and M. Vaziri, “I can parse you: Grammars for dialogs,” in *2nd Summit on Advances in Programming Languages (SNAPL 2017)*, ser. LIPIcs, vol. 71. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, pp. 6:1–6:15.
- [21] J. Porubán, M. Forgáč, and M. Sabo, “Annotation based parser generator,” in *2009 International Multiconference on Computer Science and Information Technology*, Oct. 2009, pp. 707–714.

A Comparison Study of Similarity Measures in Rough Sets Clustering

Arnold Szederjesi-Dragomir^{*‡}, Radu D. Găceanu^{*‡}, Horia F. Pop^{*‡} and Costel Sârbu^{†‡}

^{*}Department of Computer Science

[†]Department of Chemistry

[‡] Babeş-Bolyai University, Cluj-Napoca, Romania

Email: {arnold, rgaceanu, hfpop}@cs.ubbcluj.ro and csarbu@chem.ubbcluj.ro

Abstract—The proper selection of the similarity measure may be of paramount importance for the clustering result especially if the clusters overlap. Therefore, the aim of this paper is to study the influence of some widely known similarity measures on the clustering structure. The analysis uses a clustering algorithm based on rough sets which is able to discover hybrid data (outliers and instances that are close enough to more than one cluster). In this paper, we examine the similarity measures influence on the overlapping regions as well. Since the standard data sets do not offer specific information regarding the overlapping areas, we also propose an approach for extracting this data in order to be used for benchmarking purposes. Experiments conducted on standard data sets outline the importance of the proper similarity measure selection.

Index Terms—clustering, distance, rough sets, software agent

I. INTRODUCTION

The problem of clustering and classification lies in partitioning a set of data into groups, one group for each input category. It therefore attempts to include each instance into one of the possible classes or clusters. The complete description of the classes is not known in real life, although a finite and usually smaller number of instances (training set) are available, which often provide partial information for the optimal design of the classification or clustering system. Clustering is an unsupervised learning problem and it deals with partitioning an unlabeled set of data in such a way that two objects from the same group are as similar as possible, while objects from different groups are as different as possible.

The decision of grouping two instances together is based on some similarity or closeness measure and hence the proper selection of this measure is extremely important for the outcome of a cluster analysis process.

One of the major issues in real-life data analysis is uncertainty management, which includes problems like: overlapping clusters, clusters with arbitrary shape, and noisy data. In this regard, to deal with uncertainty and vagueness, the concepts of fuzzy sets [1] theory and rough sets [2] theory were applied. In a fuzzy or rough sets clustering scenario, the chosen distance measure could have an even greater impact on the outcome of a clustering process if the uncertainty regions are also analyzed.

Some studies on the similarity measure influence in clustering algorithms have been performed before ([3]–[5]), but, to our knowledge, there are no such studies for fuzzy or rough clustering scenarios, i.e., for uncertainty driven environments.

In [3], the authors perform a comprehensive study considering several distance measures and clustering algorithms aiming to help the research community in identifying suitable distance measures and to facilitate a comparison and evaluation of newly proposed similarity measures with traditional ones. In [4], the authors study the influence of similarity measure selection in conjunction with several clustering algorithms on high dimensional sparse data representing web documents. They conclude that, in this context, the Euclidean distance performs the poorest. In [5], the authors perform a comparative evaluation of a variety of similarity measures for categorical data sets observing that some measures are able to have a consistently higher performance than others on specific data sets.

In this paper, our aim is to investigate the way the clustering outcome is influenced by the distance measure selection, *when dealing with overlapping clusters*. More precisely, we would like to see the way overlapping regions change according to the selected distance measure.

The main contributions of this paper are the following:

- A comparative evaluation in terms of accuracy of some of the most widely used similarity measures
- An approach for determining the overlapping regions to be used for benchmarking
- A study of the influence of similarity measure selection on overlapping clusters

The rest paper is structured as follows: Section II presents the motivation of our study, emphasizing its importance, Section III presents a theoretical background regarding rough sets clustering. The proposed methodology is presented in Section IV, and experiments on standard data sets are detailed in Section V. Finally, conclusions and ideas for future work are presented in Section VI.

II. MOTIVATION

Pattern recognition tasks are generally designed to perform an approximate match of all possible inputs and this requires using some sort of similarity measure. Clustering algorithms are no exception to this situation and the primary goal of this paper is to study the influence of different similarity measures on the clustering result.

The problem is that data sets are structurally different from each other. For example, in some data sets classes can be

linearly separated, in other data sets there is no clear separation between classes. This is why it is important to adapt the considered similarity measure to the data set on which it is going to be applied. Considering this issue, in this paper we apply several similarity measures on standard data sets in the attempt to find the one which fits the best every data set. Similar studies have been performed before in the literature ([3]–[5]), but, as far as we know, there are no such studies for uncertainty driven environments, i.e., overlapping clusters.

Therefore we believe that besides considering the quality of a clustering solution from an accuracy standpoint, it is also highly important to analyze the way overlapping regions are affected by the similarity measure choice.

But studying the influence of different similarity measures over the overlapping regions raises an important issue: the fact that the standard data sets do not provide precise information regarding these overlapping areas which makes it difficult to compare the effects of various distance measures over these regions. This is why is also propose a methodology for extracting this kind of information form a data set in order to be used for benchmarking purposes.

III. ROUGH SETS CLUSTERING

Pattern recognition is an activity in which human beings normally excel, but due to the uncertainty involved in the whole process, it is not a trivial task for a computer program. Incomplete data, vagueness in class definitions, and the presence of outliers are sources of uncertainty and in such cases the theories of fuzzy sets and rough sets are employed [1], [2].

The rough sets theory [2] is a major mathematical instrument for managing uncertainty and it consists in offering a formal approximation of a crisp set in terms of a pair of sets representing the lower and upper bound of the original set.

Given a set of objects U called the universe of discourse, an equivalence relation $R \subseteq U \times U$, and a subset of U denoted by X , in order to approximate X with respect to R we consider the following definitions:

Definition 1: The **lower approximation** of a set X with respect to R is the set of all objects which **certainly** belong to X : $\underline{R}X = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$.

Definition 2: The **upper approximation** of a set X with respect to R is the set of all objects which **possibly** belong to X : $\overline{R}X = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \emptyset\}$.

Definition 3: The **boundary region** of a set X with respect to R is the set of all objects which can not certainly be classified as either belonging to X or not belonging to X : $RB = \overline{R}X - \underline{R}X$.

Definition 4: A **rough set** is a tuple $\langle \underline{R}X, \overline{R}X \rangle$, where $\underline{R}X$ is the **lower approximation** (Definition 1) of the target set X and $\overline{R}X$ is the **upper approximation** (Definition 2) of the target set X .

Let $\underline{R}C_k$ and $\overline{R}C_k$ denote the lower and upper approximations of a cluster C_k , and let $RB = \overline{R}C_k - \underline{R}C_k$ be the boundary region of C_k . The tuple $\langle \underline{R}C_k, \overline{R}C_k \rangle$ is called the rough set associated to C_k with respect to some equivalence

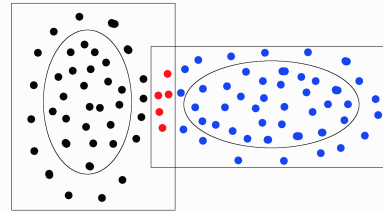


Fig. 1. Rough clusters with overlapping upper approximations [6].

relation R (see Definition 4). Then the rough sets clustering problem may be defined as in Definition 5.

Definition 5: Rough sets clustering [6] is the process of finding a set $RC = \{\langle \underline{R}C_k, \overline{R}C_k \rangle | k = \overline{1, p}\}$ of subsets of a given set of objects $X = \{x^i | i = \overline{1, n}, 1 \leq p \leq n\}$ such that:

- $\forall k, l = \overline{1, p}, k \neq l : \underline{R}C_k \cap \underline{R}C_l = \emptyset$
- $\forall k, l = \overline{1, p}, k \neq l : |\overline{R}C_k \cap \overline{R}C_l| \geq 0$
- $\bigcup_{k=1}^n \overline{R}C_k = X$
- $\forall k = \overline{1, p} : \underline{R}C_k \neq \emptyset$
- $\forall k = \overline{1, p} : \underline{R}C_k \subseteq \overline{R}C_k$
- $\forall k = \overline{1, p}, \forall i, j = \overline{1, |\underline{R}C_k|} : x^i R x^j \wedge x^i, x^j \in C_k$
- $\forall k = \overline{1, p}, \forall i, j = \overline{1, |\overline{R}C_k|} : x^i R x^j \wedge |\{x^i, x^j\} \cap C_k| \geq 0$.

So in a rough sets clustering approach, every cluster C_k is defined in terms of a lower approximation and an upper approximation. The instances from the lower approximation certainly belong to the cluster C_k with respect to some similarity measure. The instances from the upper approximation possibly belong to C_k , but we cannot be certain about this, according to the considered similarity measure. The whole idea may be visualized in Figure 1, where two not clearly separable clusters are shown. Even though they can not be precisely identified, the two clusters may still be described by their lower approximations (the ellipses) and by their upper approximations (the rectangles). In this example the boundary regions are overlapping and the red instances from the image belong to both upper approximations and hence, to some extent, to both clusters [6].

IV. METHODOLOGY

This section is devoted to explain the methodology used in this study for evaluating the effect of similarity measures on the outcome of a clustering process. This paper aims primarily to analyze the impact of different distance measures on the quality of a clustering algorithm in case of overlapping clusters.

The experiments are conducted on three standard data sets: Iris [7], Seeds [8] and Wine [9]. For each data set we evaluate the accuracy of the ABARC algorithm (from [6]), as well as the reported rough instances (instances that are close enough to more than one cluster), by selecting in turn each of the aforementioned distances. The clustering algorithm (Section IV-A) is implemented in Elixir [10], while the analysis part (Sections IV-B, IV-C and IV-D) is implemented in Ruby [11].

A. An overview of the clustering process

This section presents an overview of the main steps performed by the ABARC (Agent BAsed Rough Clustering) algorithm introduced in [6]. The ABARC algorithm addresses the overlapping clusters problem by modeling clusters using notions from the rough sets theory. It successfully identifies outliers and, by using software agents, it is also scalable. The algorithm uses the following arguments: X — the data set, i_{max} — the number of iterations, λ — the maximum number of times an agent may search for a similar one, σ_1 — the similarity limit, δ — the similarity measure. The main steps of the algorithm are as follows:

- 1) Initialize data and parameters: $X, i_{max}, \lambda, \sigma_1, \delta$.
- 2) Let \mathcal{AG} be a set of agents, where each agent is associated with one instance from the input data set X .
- 3) Each *agent* from \mathcal{AG} will *asynchronously* start to search for a *similarAgent* :
 - a) Nondeterministically select an agent from \mathcal{AG} .
 - b) If they are not in the same cluster and if their similarity is less than σ_1 then a *similarAgent* is found. *GOTO Step 4* with the *similarAgent*.
 - c) Decrement λ .
 - d) If $\lambda = 0$ then *GOTO Step 5* with no *similarAgent*.
 - e) If $\lambda > 0$ then *GOTO Step 3a*.
- 4) If a *similarAgent* is found then the *agent* moves to its cluster.
- 5) Decrement i_{max} .
- 6) If $i_{max} > 0$ then *GOTO Step 3*.
- 7) Stop.

The clustering process described above starts by initializing the input data and parameters. The data set X is normalized using Min-Max normalization [12] and each instance is associated with one agent. Also, there is one cluster for each agent so, in the beginning, the number of clusters is equal to the number of agents which is equal to the number of instances. Each agent executes in *parallel* (Step 3) and it searches for a similar one based on the provided similarity measure δ and the similarity limit σ_1 . If a *similarAgent* is found then they are grouped together (Step 4) otherwise the search process continues. After λ failed attempts the search for similar process is aborted for the current agent, leaving the task to a different agent or to another iteration (Step 5).

Since agents are grouped together only if they should *certainly* belong to the same cluster (based on the similarity limit, σ_1), the first phase of our approach will probably produce a large number of clusters. A second phase (introduced in [6]) unifies similar clusters producing *rough clusters*. In this phase, the unification process is based on a second similarity limit, σ_2 , denoting the level up to which two instances are *possibly* similar.

Even after the second phase there might be a significant number of clusters remaining, but most of them are normally composed of a very small number of entities which are not similar to either of the ‘normal’ clusters. The instances from these small clusters will be marked as possible *outliers*.

Nevertheless, in the third phase of our approach (presented in [6]), we assign them to the closest cluster and we get the final clustering structure. So the algorithm produces a set of clusters such that:

- 1) all instances from the data set belong to at least one cluster
- 2) given any two clusters their intersection might not be the empty set, i.e., the clusters might overlap; we refer to the instances from the overlapping regions as *rough instances*
- 3) even if they are assigned to the closest cluster, the outliers are clearly marked as such

B. Similarity measure evaluation based on accuracy

In order to study the influence of different similarity measures on a rough clustering process, our first approach is to compute the accuracy of the algorithm described in Section IV-A for each similarity measure.

In this study, we examine the following similarity measures: Manhattan, Chebyshev, Euclidean and Minkowski [12]–[14]. Actually, the *Minkowski* distance between two instances x and y is a generalized form of the other three distances and it is defined by $d_{MIN} = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$, where $p \in \mathbb{R}$, $p \geq 1$, and $x, y \in \mathbb{R}^n$. For $p = 1$ it corresponds to the *Manhattan* distance, while for $p = 2$ it corresponds to the *Euclidean* distance. The *Squared Euclidean* distance (which is just the square of the Euclidean distance) is also considered in our study. When p reaches infinity, the *Chebyshev* distance is obtained: $\lim_{n \rightarrow \infty} \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} = \max_{1 \leq i \leq n} |x_i - y_i|$.

We consider three standard data sets (Iris [7], Seeds [8] and Wine [9]) and, given a similarity measure, we execute the clustering algorithm several times for each data set. We collect the accuracy from each execution and we compute the minimum, maximum and average accuracy. This process is repeated for each similarity measure.

C. Identifying rough instances for benchmarking purposes

The evaluation process from Section IV-B uses the accuracy to compare similarity measures in a rough clustering context. However, a more suitable approach when dealing with hybrid data would be to somehow compare the rough instances produced by the algorithm using a certain similarity measure with the *actual* or *true* rough instances.

The accuracy of a clustering algorithm is computed based on the *actual* or *true* cluster structure given in the official data set documentation. In the same fashion, we would like to find some index (like accuracy) that measures the roughness of an approach, i.e., an indicator for the quality of the reported rough instances. Such an index should, of course, not be biased and one possible way in achieving this would be to tie its calculation to the *actual* or *true* rough instances in a similar way the accuracy does.

Unfortunately, the first problem in achieving the aforementioned goal is the fact that the official documentation of any data set (as far as we know) does not specify the actual rough

Algorithm 1: Find Rough Instances

Data: C, τ, δ
Result: RI — the set of actual rough instances

```

1  $RI = \emptyset$ 
2 while true do
3    $ID = \text{computeMeanDistances}(C, \delta)$ 
4    $RI_1 = \text{computeRoughInstances}(ID, \tau)$ 
5   if  $RI_1 = \emptyset$  then
6     break
7   end
8    $C = C \ominus RI_1$ 
9    $RI = RI \cup RI_1$ 
10 end

```

instances. In practical scenarios, the hybrid instances could be validated by a domain expert, but there should also be other possibilities, at least for benchmarking purposes. So, our first intent is to propose an objective approach for finding the *actual* or *true* rough instances from a data set and, secondly, to introduce a first attempt for a roughness index that takes into account this information.

Algorithm 1 shows the proposed process of finding the actual rough instances. The algorithm receives as a first argument the set of clusters $C = \{C_k | k = \overline{1, p}\}$, as specified in the official data set documentation, where p represents the number of clusters and C_k are subsets composed of elements from the data set $X = \{x^i | i = \overline{1, n}, 1 \leq p \leq n\}$. The second argument, τ , denotes the *rough threshold* and it is used for deciding if an instance should be considered a rough one. The last argument, δ , represents the distance function used in the analysis process. The algorithm returns the set RI of instances that will be considered truly rough.

The algorithm starts by computing a mapping, ID , between each instance and the mean distance corresponding to each cluster: $ID = X \xrightarrow{m} M$, where the \xrightarrow{m} symbol denotes a mapping between each element of X and a set $M = \{m_k | k = \overline{1, p}\}$ of mean distances. Given an instance $x^i \in X$, the mean distance corresponding to cluster C_k is: $m_k = \frac{\sum_{j=1}^{|C_k|} \delta(x^i, x^j)}{|C_k|}$, $x^j \in C_k$, $j \neq i$. In line 4 from Algorithm 1, the set RI_1 of rough instances is computed by selecting all instances from ID having the following property: $\exists i, j \ni |m_i - m_j| \leq \tau$, where $m_i, m_j \in M$ represent, respectively, the mean distances of clusters C_i and C_j for a given instance, and τ is the rough threshold. The instances from RI_1 are removed from the set of clusters, as indicated by the \ominus operator, and they are added into the result set, RI . The whole process is repeated until the set RI_1 is empty.

D. Similarity measure evaluation based on the rough instances

In order to evaluate the influence of a certain similarity measure on the rough instances, we compare the rough instances reported by the algorithm described in Section IV-A with the

rough instances proposed by the Algorithm 1 from Section IV-C.

Given a similarity measure, we execute the algorithm several times (N) on a certain data set and we collect the reported rough instances from each execution. For each rough instance x^i we compute the *occurrence rate*, $occ_{x^i} = \frac{n}{N}$, where n denotes the number of occurrences of the instance in the series of N executions.

Definition 6: The **rough score** of a given similarity measure is the ratio between the sum of the *occurrence rates* of the validated rough instances and the sum of all *occurrence rates*:

$$\mathcal{R}_{sc} = \frac{\sum_{i=1}^{|RI_d|} 1_{RI}(x^i) \cdot occ_{x^i}}{\sum_{i=1}^{|RI_d|} occ_{x^i}},$$

where: RI_d is the set of rough instances reported by the clustering algorithm for a certain similarity measure (or distance, d); RI is the set of actual rough instances produced by the Algorithm 1 from Section IV-C; 1_{RI} is the indicator function of RI : $1_{RI}(x^i) = \begin{cases} 1, & \text{if } x^i \in RI \\ 0, & \text{otherwise.} \end{cases}$

Given a similarity measure, we execute the clustering algorithm N times for each data set computing the *rough scores*. This process is repeated for each similarity measure.

V. RESULTS AND DISCUSSION

This section presents the experimental setup of our study and discusses the obtained results. The experiments are conducted on three standard data sets: Iris [7], Seeds [8] and Wine [9].

The Iris data set [7], one of the most widely used data sets in pattern recognition, contains 150 instances with four attributes denoting geometric information regarding three species of Iris plants. There are three classes with 50 instances each, one of the classes being linearly separable from the other two. The data is scaled in the $[0, 1]$ range using Min-Max normalization [12] and the algorithm described in Section IV-A is applied with the following parameter setting: $i_{max} = 100$, $\lambda = 100$, $\sigma_1 = 0.0115$, and $\sigma_2 = 0.1$. For each of the considered distances, the clustering algorithm is executed 50 times computing the accuracy and rough score as described in Section IV. Table I shows the minimal, maximal and average accuracy for each similarity measure. The rough instances together with their occurrence rate (expressed in %) are shown in Table II.

The Seeds data set [8] contains 210 instances with seven attributes describing geometric parameters of wheat kernels. There are three classes of 70 instances each. After scaling the data in the $[0, 1]$ range using Min-Max normalization [12], the algorithm from Section IV-A is applied with the following parameter setting: $i_{max} = 100$, $\lambda = 100$, $\sigma_1 = 0.024$, and $\sigma_2 = 0.2$. The clustering algorithm is executed 50 times for each of the considered similarity measures and a result summary in terms of accuracy is presented in Table I, while the obtained rough instances are shown in Table II.

The Wine data set [9] contains 178 instances with 13 attributes describing information of a chemical analysis of

wines grown in the same region in Italy but derived from three different cultivars. The data is normalized in the same fashion as for the other data sets and the algorithm described in Section IV-A is applied 50 times with the following parameter setting: $i_{max} = 100$, $\lambda = 100$, $\sigma_1 = 0.145$, and $\sigma_2 = 0.45$. Tables I and II show the results for the Wine data set (accuracy and rough instances).

The analysis process described in Sections IV-C and IV-D is then applied for each data set obtaining the rough scores displayed in Table III. In computing the rough scores, only the instances with an occurrence rate higher than 50% from Table II were taken into account. A *NaN* value in Table III indicates that no rough instances have been reported by the clustering algorithm in that case.

Considering Table I, the best results on the average are for the Minkowski distance where $p = 2.3$. From Table III, it appears that the best result is obtained for the Squared Euclidean distance. But this result is biased because the evaluation methodology presented in Section IV-C uses precisely this similarity measure. The next best result is obtained again for the Minkowski distance with $p = 2.3$.

From both Tables I and III the following order relations between the scores (accuracy, rough) of different Minkowski distances (expressed as values of p) appear to emerge: $(p = 1) \leq (p = \sqrt{2}) \leq (p = 2) \leq (p = 2.3)$ and $(p = 2.3) \geq (p = 2\sqrt{2}) \geq (p = 3) \geq (p = \infty)$, where $(p = n)$ denotes the score (accuracy or rough) of the Minkowski distance for $p = n$. The exceptions to this possible rule are the accuracies for $(p = 1)_{iris}$, $(p = 2)_{seeds}$, $(p = \infty)_{seeds}$ and the rough score for $(p = 3)_{iris}$. Also, the rough scores from the Wine data set are not taken into consideration since there are relatively few rough instances with high occurrence rates in Table II that turn out to be false positives in general leading to many rough scores of 0 (see Table III).

Even though our experiments indicate that the Minkowski distance with $p = 2.3$ is the best one, we avoid drawing this conclusion because a possible reason for this situation could be the fact that the values of p close to 2 favor clusters of spherical shape. The fact that for $p = 2.3$ we obtain the best results in all data sets could simply happen because all clusters might have similar spherical shapes in all cases. The scores tend to decrease for $p > 2.3$ probably because the value of the Minkowski distance decreases with increasing values of p and hence more instances are accepted as similar including possible outliers. Also, with increasing values of p the sphere like shape of clusters expands, becoming a square for $p = \infty$. Similarly, for decreasing values of p the spherical shapes shrink accepting fewer and fewer instances, which could explain why the scores tend to degrade for smaller values of p .

VI. CONCLUSIONS AND FUTURE WORK

One of the difficulties facing experts and researchers when trying to work with a clustering algorithm is selecting the correct distance measure because, as shown in this paper, they behave differently for the same data set. The aim of

this paper was to investigate the influence of several similarity measures on the outcome of a clustering algorithm based on the rough sets theory by studying the influence on the accuracy and on the rough instances. Investigating the influence on rough instances was challenging because the standard data sets do not offer specific information regarding the overlapping regions. This is why we have also proposed in this paper a methodology for determining possible rough instances from a data set, an approach that could be useful for benchmarking purposes. Our results indicate that the Minkowsky distance with $p = 2.3$ is the best one from both an accuracy and rough instances standpoint, but, as discussed in the paper, this might be primarily caused by the fact that all clusters from all considered data sets have similar spherical shape. As future work, we would like to perform more experiments on other data sets with larger overlapping regions and with clusters having different shapes. We would also like to study the way outliers are influenced by the similarity measure selection.

REFERENCES

- [1] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.
- [2] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Norwell, MA, USA: Kluwer Academic Publishers, 1992.
- [3] A. S. Shirshorshidi, S. R. Aghabozorgi, T. Y. Wah, and A. R. Dalby, "A comparison study on similarity and dissimilarity measures in clustering continuous data," in *PloS one*, 2015.
- [4] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in *In Workshop on Artificial Intelligence for Web Search (AAAI 2000)*. AAAI, 2000, pp. 58–64.
- [5] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *SDM*, 2008.
- [6] R. D. Găceanu, A. Szederjesi-Dragomir, H. F. Pop, and C. Sârbu, "ABARC: An agent-based rough sets clustering algorithm," 2019, unpublished.
- [7] R. A. Fisher, *UCI Machine Learning Repository: Iris Data Set*, <http://archive.ics.uci.edu/ml/datasets/Iris>, 1936. [Online]. Available: <http://archive.ics.uci.edu/ml/datasets/Iris>
- [8] P. Kulczycki, *UCI Machine Learning Repository: Seeds Data Set*, <https://archive.ics.uci.edu/ml/datasets/seeds>, 2012. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/seeds>
- [9] M. Forina, *UCI Machine Learning Repository: Wine Data Set*, <https://archive.ics.uci.edu/ml/datasets/wine>, 1991. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/wine>
- [10] P. L. Elixir, "Elixir," 2019. [Online]. Available: <https://elixir-lang.org/>
- [11] P. L. Ruby, "Ruby," 2019. [Online]. Available: <https://www.ruby-lang.org/en/>
- [12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
- [13] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007.
- [14] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," 2007.

TABLE I
ACCURACIES FOR THE IRIS, SEEDS AND WINE DATA SETS

Similarity measure	Accuracy Iris (%)			Accuracy Seeds (%)			Accuracy Wine (%)		
	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Manhattan	86.0	95.69	96.67	61.43	86.17	87.62	78.65	88.33	91.01
Chebyshev	83.33	84.07	85.33	83.33	90.36	90.95	89.89	91.52	92.7
Euclidean	91.33	96.49	96.67	79.52	91.1	91.43	72.47	96.81	98.31
Squared Euclidean	88.0	96.36	96.67	90.95	90.98	91.43	72.47	96.57	97.75
Minkowski $p = \sqrt{2}$	85.33	96.39	97.33	89.05	91.32	91.43	82.02	92.07	92.7
Minkowski $p = 2.3$	96.0	96.65	96.67	90.95	91.39	91.43	84.83	97.48	97.75
Minkowski $p = 2\sqrt{2}$	96.0	96.65	96.67	89.52	90.97	91.9	84.83	95.34	96.63
Minkowski $p = 3$	94.0	96.6	96.67	73.81	90.29	91.9	78.65	92.62	95.51

TABLE II
ROUGH INSTANCES FOR THE IRIS, SEEDS AND WINE DATA SETS

	Manhattan		Chebyshev		Euclidean		Squared Euclidean		Minkowski $p = \sqrt{2}$		Minkowski $p = 2.3$		Minkowski $p = 2\sqrt{2}$		Minkowski $p = 3$	
	I#	%	I#	%	I#	%	I#	%	I#	%	I#	%	I#	%	I#	%
Iris data set	94	60	15	100	15	100	60	98	15	100	135	86	60	98	60	96
	61	60	74	98	135	90	135	80	135	92	60	74	135	88	135	82
	58	60	129	70	109	64	107	40	109	52	109	58	107	74	107	66
	60	42	133	70	136	44	99	38	136	48	107	54	109	36	99	38
	118	12	60	66	122	8	150	38	60	6	136	46	99	34	94	38
	130	10	16	60	143	8	94	38	103	6	99	20	94	34	58	38
	136	10	120	46	60	8	58	38	105	4	94	20	58	34	109	30
	109	10	68	44	114	8	138	34	139	4	58	20	136	24	136	24
	126	10	118	34	107	8	104	34	141	4	149	10	122	20	137	20
119	10	111	30	102	8	147	34	145	4	137	10	143	20	116	20	
Seeds data set	0	0	113	76	141	68	133	86	25	100	193	70	193	100	193	98
	0	0	89	42	143	68	193	62	133	80	206	70	40	100	25	98
	0	0	193	38	185	68	206	62	60	76	133	64	133	94	40	98
	0	0	44	38	133	60	60	60	141	64	40	60	60	70	163	86
	0	0	101	38	60	54	190	56	143	64	60	54	129	52	133	70
	0	0	82	32	190	50	40	48	185	64	76	48	13	52	113	52
	0	0	94	32	193	48	189	46	13	62	72	48	84	52	129	50
	0	0	189	30	40	48	13	44	89	56	123	48	198	52	13	50
	0	0	136	26	206	48	101	40	193	52	134	48	58	50	84	50
	0	0	99	24	76	40	105	38	82	52	58	42	50	42	198	48
Wine data set	0	0	113	100	15	88	77	66	8	84	94	80	77	70	94	88
	0	0	94	80	94	60	94	32	20	74	77	76	176	34	176	48
	0	0	133	72	64	46	177	32	3	74	8	40	150	34	150	48
	0	0	141	42	99	46	8	30	56	74	15	38	94	34	120	32
	0	0	143	42	77	36	47	30	29	74	14	38	136	24	22	30
	0	0	162	42	47	34	44	24	46	74	4	38	177	24	80	30
	0	0	163	42	177	26	42	22	77	64	161	32	135	24	77	26
	0	0	137	38	130	22	95	22	150	56	165	32	178	18	104	22
	0	0	177	38	176	22	101	20	129	54	175	32	139	18	177	22
	0	0	83	24	137	22	64	20	118	54	156	32	141	18	95	22

TABLE III
ROUGH SCORES FOR DIFFERENT SIMILARITY MEASURES

Similarity measure	Rough score		
	Iris	Seeds	Wine
Manhattan	0.33	NaN	NaN
Chebyshev	0.14	0	0.39
Euclidean	0.6	0.35	0
Squared Euclidean	1	0.63	0
Minkowski $p = \sqrt{2}$	0.59	0.26	0.26
Minkowski $p = 2.3$	0.8	0.59	0
Minkowski $p = 2\sqrt{2}$	0.71	0.37	0
Minkowski $p = 3$	0.72	0.13	0

Coordinated jogging algorithm for general serial robot architectures

Martin Švejda* Tomáš Myslivec* Martin Goubej*

* *University of West Bohemia, NTIS Research Centre, Pilsen, Czech Republic (e-mail: msvejda@ntis.zcu.cz, tmyslive@ntis.zcu.cz, mgoubej@ntis.zcu.cz).*

Abstract: The paper deals with a novel functional block for general serial robot coordinate jogging. The functional block is implemented and integrated to the functional block library of the real-time control system REXYGEN and brings new possibilities of rapid prototyping and initial testing of advanced robot architectures. The robot kinematics is described by the Denavit-Hartenberg parameters. The common jogging mode of operation (world, tool) is supported as well as user-defined coordinate frames definition for specific applications. In the case of kinematic redundancy, the internal robot motion optimization is available allowing joint position limitation and/or obstacle avoidance. Maximum allowed joint velocity is taken into account in order to cope with the kinematic singularities. Illustrative examples of the early-prototype robot for industrial inspection purposes are introduced and universality of the proposed jogging functional block is verified.

Keywords: Industrial robot; rapid prototyping; coordinate jogging; REXYGEN control system; motion optimization; obstacle and limitation avoiding;

1. INTRODUCTION

Industrial robotics employs a large number of kinematic architectures ranging from simple ones like planar and SCARA robots or standard industrial arms/manipulators (serial 6 DoF robot with spherical wrist) to advanced and complex concepts like parallel or hybrid architectures, robots for special applications and experimental research prototypes. The robot controllers are often made to fit a specific robot type and it is supposed that the robot structure will not change except for standard setting options like tool and/or base compensation, etc. Especially in the case of research activities concerning new robot architectures development, it is necessary to have a possibility to test early robot prototypes in the manner of coordinate jogging in the base, tool or other specified coordinate systems (CS). In addition, the commercial controllers are closed systems where advanced and experimental features including e.g. redundancy resolution for optimal robot motion, robot kinematic reconfiguration or joint position/velocity/acceleration constraints can not be implemented in a simple way. In the current trend of rapid prototyping the tools for fast and efficient design of new robot architectures play crucial role in the field of robotics.

One of the possibilities to deal with the motion control of multi-axis mechatronic systems of non-standard user-defined kinematic architectures is well known Programmable Multi-Axis Controller (PMAC). PMACs are controls for which the entire hardware and software structure is organized around managing complex motion. The example of the PMAC is e.g. OMRON CK3E Series [12] with up to 32 controlled axes for advanced motion and CNC machining with G-code capability. The for-

ward/inverse kinematic algorithm can be managed via matrix handling and the space conversion which makes possible e.g. complex kinematic control of hexapod telescope mirror.

Many software packages are devoted to multi-axis control of mechatronic systems including standard industrial robot configurations as well as user-defined general kinematic architectures. *Energid technologies corporation* offers highly sophisticated motion control for industrial, medical, commercial, collaborative and consumer robotic systems. Their *Actin SDK* [18] introduces "Real-Time Adaptive Motion Control software for Any Robot". *Actin SDK* offers possibility to define general kinematics and dynamics model, optimization of the motion (joint limit and singularity avoidance). *Energid Technologies Corporation* has been funded by NASA and *Actin 5 SDK* is labeled as software which brings enabling tools and capability for NASA space applications [3]. *MoveIt* motion planning framework [11] enables to incorporate latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation. Over 100 robot kinematic architectures are included and users are allowed to write their own inverse kinematics algorithms (open/closed/branch kinematic loops) using *KLD* (numerical Jacobian-based solver: <http://www.orocos.org/kdl>). *IKFast Plugin* makes possible to implement own kinematics solvers. Some relevant applications are addressed to mobile manipulation task simulation [5], pick and place application with regards to given constraints (joints limit, obstacles, etc.) [7] and AUV control system implementation [21]. *RoboDK* [16] simulation environment brings offline robot programming features like optimization of robot path, avoiding singularities, axis limits and collisions overcoming. Moreover the trajectory post proces-

sor for a specific robot controller (ABB, FANUC, KUKA, MOTOMAN, etc.) is available in order to upload the program to native programming language. *RoboAnalyzer* [6; 15] is a 3D model based software that can be used to teach and learn the robotics concepts. It includes 3D model generator based on Denavit-Hartenberg parameters for serial manipulators with revolute and prismatic joints as well as functions for joint/cartesian level jogging, visualization of the robot movement, etc. *Virtual Robot Module* as part of the *RoboAnalyzer* can be integrated with Matlab.

Despite the range of advanced features provided by the above-mentioned tools, their integration into industrial grade environment remains difficult. Most of the generic PLC-based Motion Control (MC) systems rely on the PLCOpen Motion Control standard defining a functional block library implementing commonly required functionalities (PLCOpen Motion Control, see <https://www.plcopen.org/technical-activities/motion-control>). The Coordinated Motion part of the standard covers both motion instructions (linear, circular and generic smooth path interpolation) and kinematic transforms. However, one of its main drawbacks is a lack of support for rapid prototyping of new robot architectures without time-consuming re-implementation of the forward/inverse kinematics whenever some change in the robot structure occurs. This includes also the functionality of the generic jog allowing to move the robot in a coordinated manner in a defined direction and a chosen coordinate reference system.

The goal of the paper is to propose an enhancement of the existing industrial MC blockset which implements a universal coordinated (Cartesian) robot jogging for a general serial robot kinematics with revolute joints. The key functionality requirements are given as follows:

- Arbitrary robot architectures with revolute joints (e.g. number of joints, rotation axis arrangement, etc.)
- Base or/and end-effector (tool) compensation
- End-effector (Cartesian) jogging including user-defined CSs reference (e.g. jog in specific plane only, etc.)
- Joints position limits detection and stopping the end-effector motion if the limits are reached
- Proximity to singularity detection and stopping the end-effector motion, further movement is allowed only in the directions going away from the singularity
- Limitation the maximum joint speed (no robot joints can rotate faster than given limit at the expense of slowing down the end-effector coordinated motion speed)
- Redundancy resolution (more joints than controlled end-effector DoFs) in the sense of joint position limit overcoming and/or user-defined objective function optimization

The proposed algorithm is implemented and experimentally validated by means of REXYGEN [17] real-time control environment and two examples of unconventional industrial robot designs.

2. THEORETICAL BACKGROUND

There are many notations for robot kinematic description, but the well known are probably Denavit-Hartenberg [4] and Khalil-Kleifinger [10] notations. In order to modelling only serial kinematic chains the best known Denavit-Hartenberg (D-H) notation is sufficient. The kinematics of the robot with n revolute joints is described by $3 \cdot n$ kinematic parameters d_i , a_i , α_i for $i = 1 \dots n$ and n joint coordinates θ_i . The meaning of the kinematic parameters is depicted in Fig 1. Compensation of the base ($\star = b$) and end-effector ($\star = e$) position are given by the parameters x_\star , y_\star , z_\star (translation), γ_\star , β_\star , α_\star (ZYX Euler angles for orientation).

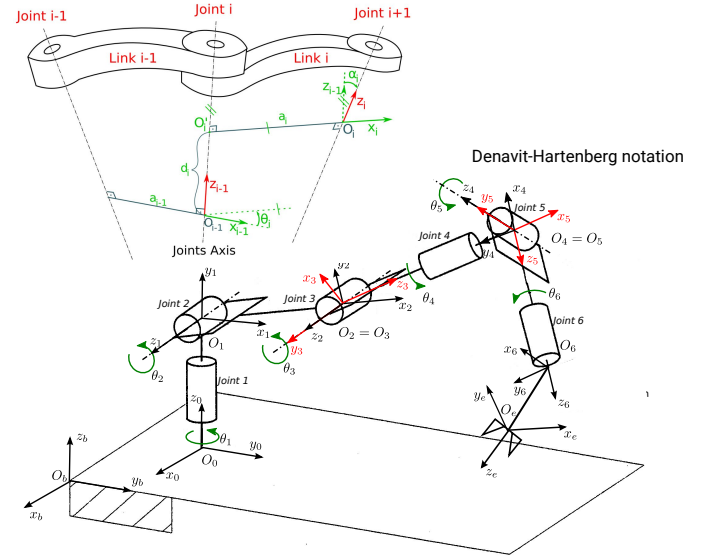


Fig. 1. Denavit-Hartenberg notation and its demonstration on 6 DoF serial industrial robot architecture

Note that the forward kinematics algorithm is given by successive multiplications of the homogeneous transformation matrices in the following form:

$$\mathbf{T}_e^b = \mathbf{T}_0^b \cdot \prod_{i=1}^n \mathbf{T}_i^{i-1} \cdot \mathbf{T}_e^n, \quad \mathbf{T}_i^{i-1} = \begin{bmatrix} \mathbf{R}_i^{i-1} & \mathbf{r}_{i-1,i}^{i-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad (1)$$

where \mathbf{R}_i^{i-1} is rotation matrix and $\mathbf{r}_{i-1,i}^{i-1}$ is translation vector between two consecutive CSs, base resp. end-effector compensation are defined in a similar manner considering unique transformation from Euler angles to rotation matrix. A subscript indicates corresponding CS and a superscript indicates the reference CS. The homogeneous transformation matrices are defined according to D-H notation and they are depending on kinematic parameters and joint coordinates as follows:

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

2.1 Robot coordinated jogging problem

Robot coordinated jogging problem is defined as finding corresponding joint velocity for a given end-effector velocity with regards to a chosen CS (e.g. world, tool, etc.). The basic dependencies between joint and end-effector velocity are:

$$\dot{\mathbf{X}}_e^b = \mathbf{J}_e^b \cdot \dot{\mathbf{Q}}, \quad \dot{\mathbf{X}}_e^b = \begin{bmatrix} \dot{\mathbf{O}}_e^b \\ \boldsymbol{\omega}_e^b \end{bmatrix}, \quad (3)$$

where $\dot{\mathbf{O}}_e^b$ resp. $\boldsymbol{\omega}_e^b$ is translation resp. angular velocity of the end-effector CS with respect the base CS, $\dot{\mathbf{Q}}$ are joint velocity and \mathbf{J}_e^b is kinematic Jacobian depending on joint position which can be generally and systematically computed from D-H kinematic parameters, see [13; 19].

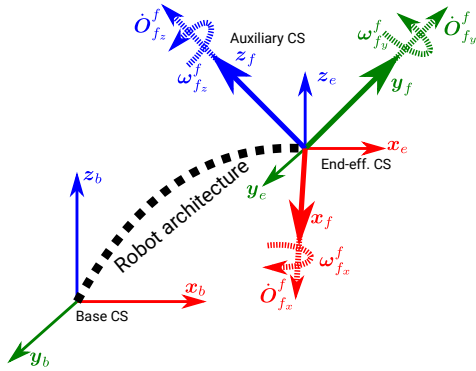


Fig. 2. Auxiliary end-effector CS definition

In order to reformulate the jogging of the robot end-effector CS F_e with respect to base CS F_b to general jogging scheme with respect to user-defined CS we define the auxiliary CS F_f , see Fig. 2. The equation (3) results in:

$$\dot{\mathbf{X}}_f^b = \begin{bmatrix} {}^T\mathbf{R}_f^b & \mathbf{r}2\mathbf{t}_f^b \\ \mathbf{t}2\mathbf{r}_f^b & {}^R\mathbf{R}_f^b \end{bmatrix} \cdot \dot{\mathbf{X}}_f^f, \quad (4)$$

where

$$\dot{\mathbf{X}}_f^f = \begin{bmatrix} \dot{\mathbf{O}}_f^f \\ \boldsymbol{\omega}_f^f \end{bmatrix}, \quad \dot{\mathbf{X}}_f^b = \begin{bmatrix} \dot{\mathbf{O}}_f^b \\ \boldsymbol{\omega}_f^b \end{bmatrix}.$$

Because the new CS F_f is located in the origin of the end-effector CS F_e (CSs are only oriented to each other), the corresponding kinematic Jacobians are identical:

$$\mathbf{J}_f^b = \mathbf{J}_e^b. \quad (5)$$

Therefore it holds:

$$\begin{aligned} \dot{\mathbf{X}}_f^b &= \mathbf{J}_f^b \cdot \dot{\mathbf{Q}}, \\ \begin{bmatrix} {}^T\mathbf{R}_f^b & \mathbf{r}2\mathbf{t}_f^b \\ \mathbf{t}2\mathbf{r}_f^b & {}^R\mathbf{R}_f^b \end{bmatrix} \cdot \dot{\mathbf{X}}_f^f &= \mathbf{J}_f^b \cdot \dot{\mathbf{Q}}, \\ \dot{\mathbf{X}}_f^f &= \underbrace{\begin{bmatrix} {}^T\mathbf{R}_f^b & \mathbf{r}2\mathbf{t}_f^b \\ \mathbf{t}2\mathbf{r}_f^b & {}^R\mathbf{R}_f^b \end{bmatrix}^{-1}}_{\mathbf{J}_f^f} \cdot \mathbf{J}_f^b \cdot \dot{\mathbf{Q}}, \quad (6) \\ \dot{\mathbf{Q}} &= (\mathbf{J}_f^f)^{-1} \cdot \dot{\mathbf{X}}_f^f, \quad (7) \end{aligned}$$

where $\dot{\mathbf{X}}_f^f$ is expected input of end-effector jogging velocity (velocity of the CS F_f with respect to the same CS), \mathbf{J}_f^f is a

new kinematic Jacobian and $\dot{\mathbf{Q}}$ is generated joint velocity. The rotation matrix ${}^T\mathbf{R}_f^b$ resp. ${}^R\mathbf{R}_f^b$ defines demanded end-effector jogging direction for translation resp. orientation and $\mathbf{r}2\mathbf{t}_f^b$ resp. $\mathbf{t}2\mathbf{r}_f^b$ are cross dependencies between demanded end-effector jogging direction for translation resp. orientation. The examples of common jogging mode can be summarized as follows:

- Word (base) jogging (jog in base CS)

$${}^T\mathbf{R}_f^b = {}^R\mathbf{R}_f^b = \mathbf{I}_{3 \times 3}, \quad \mathbf{r}2\mathbf{t}_f^b = \mathbf{t}2\mathbf{r}_f^b = \mathbf{0}_{3 \times 3}$$
- Tool (end-effector) jogging (jog in end-effector CS)

$${}^T\mathbf{R}_f^b = {}^R\mathbf{R}_f^b = \mathbf{R}_e^b, \quad \mathbf{r}2\mathbf{t}_f^b = \mathbf{t}2\mathbf{r}_f^b = \mathbf{0}_{3 \times 3}$$
 where the rotation matrix \mathbf{R}_e^b is the actual orientation of the end-effector CS.
- Word (base) translation jogging and tool (end-effector) orientation jogging

$${}^T\mathbf{R}_f^b = \mathbf{I}_{3 \times 3}, \quad {}^R\mathbf{R}_f^b = \mathbf{R}_e^b, \quad \mathbf{r}2\mathbf{t}_f^b = \mathbf{t}2\mathbf{r}_f^b = \mathbf{0}_{3 \times 3}$$

2.2 Redundancy resolution

The unique solution of the equation (6) exists only if the robot architecture is non-redundant. In the case that robot has more independent joints than DoFs of the end-effector, it is called redundant. Although kinematic redundancy is more often associated with the robots with more than six joints, the same case occurs for six axis robots if the motion of the end-effector is not fully controlled (e.g. only translation control for 6-axis industrial robot). While the number of controlled robot joints corresponds to the number of columns of the Jacobian, the number of controlled end-effector DoFs corresponds to the number of rows. The redundancy can be used for optimization of the robot (internal) motion resulting in maximization of defined joint position dependent objective function. Therefore, the optimal coordinate jog (6) is reformulated through additional term (which is projected to the the Jacobian null space) which generates internal robot joint velocity without the end-effector motion. For kinematic redundancy, the joint velocity is computed as follows, see [9; 13; 14]:

$$\dot{\mathbf{Q}} = (\mathbf{J}_f^f)^\dagger \cdot \mathbf{X}_f^f + \underbrace{\left(\mathbf{I} - (\mathbf{J}_f^f)^\dagger \cdot \mathbf{J}_f^f \right)}_{\text{Optimization term}} \cdot \frac{\partial w}{\partial \mathbf{Q}}, \quad (8)$$

where $\frac{\partial w}{\partial \mathbf{Q}}$ represents gradient of the joint dependent objective function $w(\mathbf{Q})$ to be maximized. The objective function is defined for the above mentioned key functionality requirements as follows:

- Minimization of the joint velocity (corresponding with singularity avoidance)

$$w(\mathbf{Q}) = 0. \quad (9)$$

- Joint position limit overcoming

$$w(\mathbf{Q}) = -\frac{1}{2n} \sum_{i=1}^n \left(\frac{q_i - \bar{q}_i}{q_i^{\max} - q_i^{\min}} \right)^2, \quad (10)$$

$$\bar{q}_i = q_i^{\min} + \frac{q_i^{\max} - q_i^{\min}}{2},$$

where $\mathbf{Q} = [q_1, q_2, \dots, q_n]^T$ and q_i^{\min} , q_i^{\max} define the joint position range of n robot actuators.

- Obstacles avoidance, see the example below.

2.3 Joint velocity limitation

The joint velocity limitation algorithm is based on the linear dependency (6) between joint and end-effector velocity. In the case of kinematic redundancy the additional joint velocity term resulting from optimization (8) belongs to the Jacobian null space and does not contribute to the end-effector velocity. Therefore the multiplication of the equation for the joint velocity (7, 8) by the constant factor

$$k = \max \left[1, \frac{jV_{\max}}{\max_{i=1\dots n} \|q_i\|} \right], \quad \begin{aligned} \dot{Q} &\rightarrow k \cdot \dot{Q} \\ \dot{X}_f &\rightarrow k \cdot \dot{X}_f \end{aligned}, \quad (11)$$

results in limitation of the joint velocity to the allowed maximum value jV_{\max} . For $k < 1$ the end-effector motion remains in demanded (jogging) direction but it will be slowed-down. If the constant factor is less than the allowed minimum $k < k_{\min}$, (corresponding to maximum reduction of the end-effector speed) the robot motion is stopped.

If the demanded jogging velocity \dot{X}_f is changed, the prediction to the next computation step (next time sample of the control executive) is computed and if the predicted factor $k > k_{\min}$ the robot motion is re-enabled. Therefore, the proposed algorithm allows the robot to get out of the singularity in any direction that moves away from it.

2.4 Joint position limitation

A simple algorithm is implemented which stops the robot motion ($k = 0$) ones of the joints enter to the position limit. The motion is re-enabled in the similar manner as above by comparing the actual and predicted robot joint position.

3. IMPLEMENTATION OF THE NEW GENERAL JOGGING BLOCK TO THE REXYGEN LIBRARY

REXYGEN real-time control system, see [17], is based on the programming without hand coding using the libraries of the functional blocks (e.g. similar to Matlab/Simuling programming). The general jogging block generates the demanded joint velocity depending on the current robot state (joint position) and demanded end-effector velocity (jogging). Implementation of the new

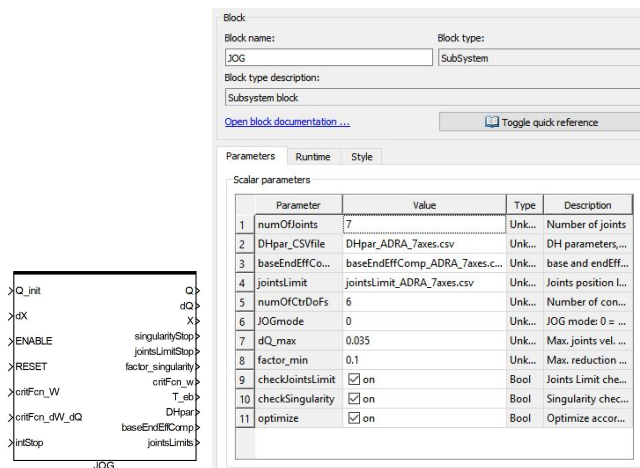


Fig. 3. General jogging functional block

functional block JOG, see Fig. 3, is based on the user programmable block REXLANG (<https://www.rexygen.com/doc/ENGLISH/MANUALS/BRef/REXLANG.html>) which implements an user-defined algorithm written in a scripting language very similar to the C language (or Java). All the above mentioned algorithms (Jacobian J_e^b and user defined-transformation matrix P computation, redundancy resolution and joint velocity and position limitation) are implemented via the REXLANG functional block. The inputs, outputs and configuration parameters are summarized as follows:

Inputs: `Q_init` initial position of the joints for integration; `dX` end-effector jogging velocity; `ENABLE` enable block; `RESET` reset block internal variables; `critFcn.W` resp. `critFcn.dW.dQ` value of objective function w resp. its gradient $\frac{\partial w}{\partial Q}$ (computed in other task, see bellow); `intStop` stop integrating (for testing purposes only);

Outputs: `Q` resp. `dQ` demanded robot joint position resp. velocity; `X` actual robot end-effector position; `singularity Stop` if true robot enters the nearly singularity position; `jointsLimitStop` nonzero value indicates corresponding joint position limitation; `factor_singularity` is factor k ; `T_eb` is matrix T_e^b ; `DHpar` D-H parameters table; `baseEndEffComp` base and end-effector compensation table; `jointsLimits` joint position limitation table;

Parameters: `numOfJoints` number of robot joints; `DHpar_CSFile` D-H parameters cfg file name; `baseEndEffComp_CSFile` base, end-effector compensation cfg file name; `jointsLimit` joint position limit cfg file; `numOfCtrDoFs` number of controlled end-effector DoFs (number of accepted rows of jacobian J_f^f); `JOGmode` mode of the end-effector jogging (world, tool or user-specified); `dQ_max` maximum joint speed; `factor_min` threshold k_{\min} , max. end-effector speed reduction; `checkJointsLimit` enable joint position limit checking; `checkSingularity` enable singularity checking; `optimize` enable optimization according to external defined objective function (else $w = 0$);

For `optimize = true` the external objective function is accepted and it is computed in separate task (typically with slow sample time) via the functional block `critFcn`, see Fig. 4. The block is implemented for two types of motion optimization: 1) Joint position limit overcoming, 2) Obstacle avoidance (obstacles are defined as cylinder surfaces with given axis, diameter, an infinite height and weight factors of the given obstacle for each robot link to avoid the obstacle).

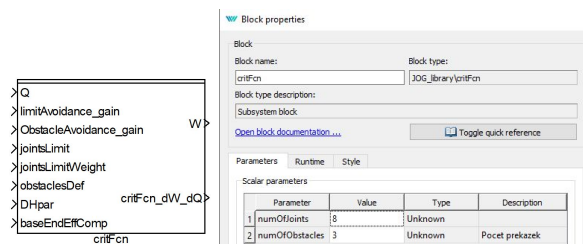


Fig. 4. Objective function computation functional block

The inputs, outputs and parameters are as follows (without those mentioned above):

Inputs: `limitAvoidance_gain` resp. `ObstacleAvoidance_gain` adjust the trade-off between the optimization modes; `jointsLimitWeight` defines the weights for joint position limit overcoming; `obstacleDef` definition of the obstacles parameters (see above);

Parameters: `fileName` user-defined REXLANG source code for objective function computation; `numOfObstacles` number of cylinder obstacles;

4. CONCLUSION AND ILLUSTRATIVE EXAMPLES

Two inspection robot prototypes ROBIN 1 and ROBIN 2 (ROBotic INSpection) following the inspection robot research summarized in [8] are considered as illustrative example of benefits of the new JOG functional block for rapid prototyping of the non-standard robot architectures. The prototyping process of the robot includes a number of modifications in the architecture (number and geometry of the joints/links, specific jogging mode, etc.) where the basic robot coordinate motion properties (jogging) has to be tested every time.

4.1 ROBIN 1

The ROBIN 1 robot is a modular tiny robot for Non Destructive Testing applications where each module consists of two mutually perpendicular revolute axes (actuated by spindle drives or belt driven actuators alternatively [1]), see Fig. 5. We consider three modules which results in 6-axis robot. The main aim of the robot is positioning of the laser pointer spot (LPS) on defined projection plane given by the point O_r , normal vector z_r and direction x_r . The position of the LPS on the plane is dependent on translation and orientation of the robot end-effector. Regarding dependency (6) we suppose substitution $\dot{X}_f^f = \dot{X}_{las}^r$ (translation velocity in XY plane with respect to plane CS) and after some computations it can be shown that the user-defined matrix P for coordinate jogging in the user defined CS will have the following form:

$$P = \begin{bmatrix} {}^T R_f^b & r 2t_f^b \\ t 2r_f^b & R_f^b \end{bmatrix} = \begin{bmatrix} (R_r^b)^T \cdot N \\ O_3 \end{bmatrix},$$

where $R_r^b = [x_r^b \ y_r^b \ z_r^b]$ is the known orientation of the projection plane and

$$N = z_e^b \cdot M + [I_{3 \times 3} | -K \cdot S(z_e^b)],$$

$$M = \frac{1}{(z_r^{bT} z_e^b)^2} \cdot \left[-z_r^{bT} \cdot (z_r^b z_e^b) | -z_r^{bT} \cdot (O_r^b - O_e^b) z_r^{bT} \right].$$

$$\cdot \begin{bmatrix} I_{3 \times 3} & O_{3 \times 3} \\ O_{3 \times 3} & -S(z_e^b) \end{bmatrix},$$

$$K = \frac{z_r^{bT} \cdot (O_r^b - O_e^b)}{z_r^{bT} z_e^b},$$

where $S(\star)$ denotes skew-matrix generated by the vector and O_e^b resp. z_e^b are the translation resp. z-axis of orientation of the end-effector given by homogeneous transformation matrix T_e^b . Only the joint position limit overcoming was used to resolve the redundancy of the robot (six controlled actuator axes and only two controlled end-effector DoFs). That makes possible to handle a large workspace despite the joints position range is strictly limited due to mechanical construction of the robot actuators.

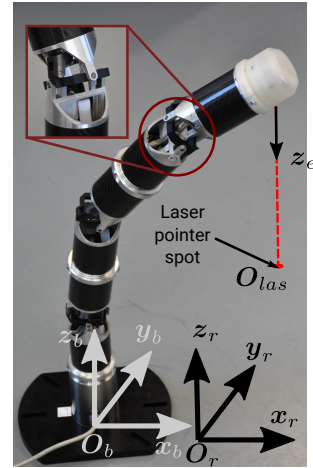


Fig. 5. ROBIN 1

4.2 ROBIN 2

The ROBIN 2 robot is a prototype of the experimental robot for inspection of pipeline welds in highly restricted areas, see Fig. 6. It consists of 1 DoF circumferential travel, 6 DoF serial chain and 1 DoF orientating (heading) of the test probe. The test probe is mounted via universal joint and it is supposed to be moved on the pipe surface with prescribed heading which results in 4 controlled DoFs of the end-effector (translation along and around the pipe and away from the pipe surface and rotation of the probe).

The user-defined matrix P for coordinate jogging on the

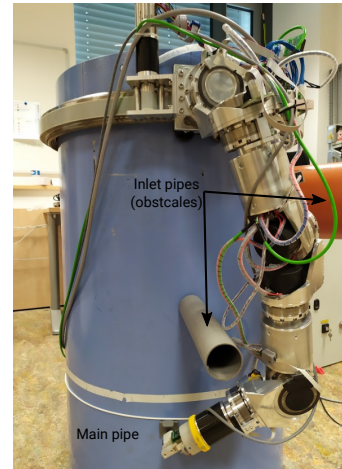


Fig. 6. ROBIN 2

pipe is simply defined by the transformation of Cartesian end-effector coordinates to cylindrical coordinates and its derivation is not further shown for simplification. The main challenge is to cope with the obstacle avoidance where the obstacles are supposed to be so called inlet pipes which enter the main pipe. Therefore, the obstacles can be described as a cylinder surface of given axis, center point and diameter. In order to implement the objective function inside the functional block `critFcn`, the obstacles will be projected to a so-called developed view, see Fig. 7. Mathematical background of the objective function computation is based on the formulation of minimum distance of the robot links and their extremal points (typically joint centers) from the cylinders axes.

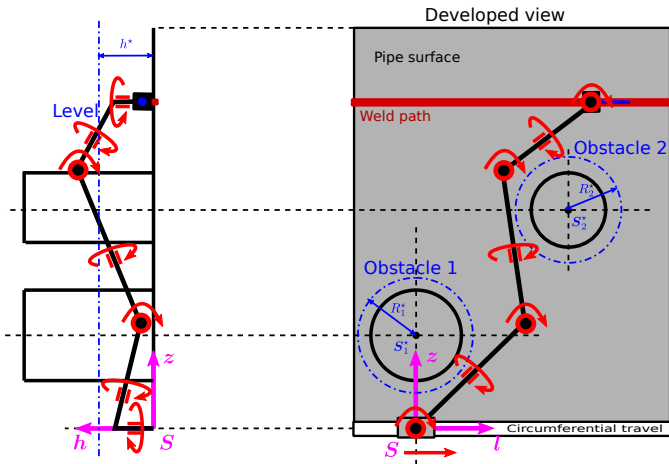


Fig. 7. Obstacles definition in the developed view

More information is given in [20; 2]. Demonstration of the end-effector jogging with obstacle avoidance is available at: <https://drive.google.com/file/d/1-fn98ad-nHwpLPRQWruKe9vBh850-H3r/view?usp=sharing>

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Industry and Trade under Grant No. FV10044.

REFERENCES

- [1] L. Bláha. Design of a compact 2-DoF joint with belt driven actuators. In *2017 IEEE 4th International Conference on Soft Computing Machine Intelligence (ISCMI)*, pages 198–202, Nov 2017. doi: 10.1109/ISCMI.2017.8279626.
- [2] L. Bláha and M. Švejda. Path planning of hyper-redundant manipulator in developed view. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 295–300, May 2018. doi: 10.1109/CarpathianCC.2018.8399644.
- [3] D. Comstock, D. Lockney, and C. Glass. A structure for capturing quantitative benefits from the transfer of space and aeronautics technology. In *International Astronautical Congress, Cape Town, South Africa*, October 2010.
- [4] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955. URL <http://ci.nii.ac.jp/naid/10008019314/en/>.
- [5] H. Deng, J. Xiong, and Z. Xia. Mobile manipulation task simulation using ros with moveit. In *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 612–616, July 2017. doi: 10.1109/RCAR.2017.8311930.
- [6] V. Gupta, R. Chittawadigi, and K. Saha, S. RoboAnalyzer: Robot visualization software for robot technicians. In *Proceedings of the Advances in Robotics, AIR '17*, pages 26:1–26:5, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5294-9. doi: 10.1145/3132446.3134890. URL <http://doi.acm.org/10.1145/3132446.3134890>.
- [7] S. Hernandez-Mendez, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez. Design and implementation of a robotic arm using ros and moveit! In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6, Nov 2017. doi: 10.1109/ROPEC.2017.8261666.
- [8] A. Jáger, T. Čechura, and M. Švejda. Non-standard robots for NDT of pipe welds. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 196–200, May 2018. doi: 10.1109/CarpathianCC.2018.8399627.
- [9] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Kogan Page Science paper edition. Elsevier Science, 2004. ISBN 9780080536613. URL <http://books.google.cz/books?id=nyrY0Pu5k10C>.
- [10] W. Khalil and J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1174–1179, Apr 1986. doi: 10.1109/ROBOT.1986.1087552.
- [11] MoveIt. *MoveIt Motion Planning Framework*, 2019. URL <https://moveit.ros.org/>.
- [12] Omron. Programmable multi axis controller, CK3E Series, NY51-A Series, 2019.
- [13] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing. Springer London, 2000. ISBN 9781852332211. URL <http://books.google.fr/books?id=v9PLbcYd9aUC>.
- [14] B. Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212, 1990. ISSN 0921-0296. doi: 10.1007/BF00126069. URL <http://dx.doi.org/10.1007/BF00126069>.
- [15] S. S. Sinha, R. G. Chittawadigi, and S. K. Saha. In *The 5th Joint International Conference on Multibody System Dynamics (IMSD2018), Lisbon, Portugal*.
- [16] RoboDK Software S.L. *RoboDK simulator for industrial robots*, 2019. URL <https://robodk.com/>.
- [17] REX Controls s.r.o. *REXYGEN (Programming Automation Devices without Hand Coding)*, 2019. URL www.rexygen.com.
- [18] Energid Technologies. *Actin, Software for Robotics Simulation and Control*, 2019. URL www.energid.com/actin.
- [19] M. Švejda. *Optimization of robot architectures (in Czech)*. PhD thesis, University of West Bohemia, 2016. URL http://home.zcu.cz/~msvejda/_publications/2016/4_SvejdaMartin_thesis_2016_06_14.pdf.
- [20] M. Švejda. Virtual simulation models for ADRA-2I NDT robot. Technical report, University of West Bohemia, 2018. URL http://home.zcu.cz/~msvejda/_publications/2018/02_ROBIN_virtSimModel.pdf.
- [21] D. Youakim, P. Ridao, N. Palomeras, F. Spadafora, D. Ribas, and M. Muzzupappa. MoveIt!: autonomous underwater free-floating manipulation. *IEEE Robotics Automation Magazine*, 24(3):41–51, Sep. 2017. ISSN 1070-9932. doi: 10.1109/MRA.2016.2636369.

Hierarchical Clustering based on IndoorGML Document

1st Judit Tamas

Eszterhazy Karoly University
tamas.judit@uni-eszterhazy.hu

Abstract—Grouping of symbolic positions using hierarchical clustering based on IndoorGML document is presented in this paper. Using hierarchical clustering information of symbolic positions, the accuracy of symbolic indoor positioning algorithms can be improved in the case of low confidence level. The symbolic positions of the physical space can be described using IndoorGML standard. Hierarchical grouping can be determined among symbolic positions by heuristic. Although it requires knowledge gained from firsthand experience. Moreover, existing hierarchical clustering algorithms can be applied using the topological description to generate tree structures. In every tree, the leaf nodes are the symbolic positions. In this paper, these generated trees are examined.

Index Terms—indoorgml, hierarchical clustering, symbolic indoor positioning

I. INTRODUCTION

Indoor positioning systems (IPS) are used to determine the position of people or objects in buildings and closed areas. IPS have been considered as an active research field since the early 1990s. However, the topic gained popularity in the last decade with the widespread of smart phones. Smart environments like homes or factories can be upgraded using indoor positioning and navigation services. In addition, IPS attracted business and the software industry. For example, the model of big, frequently visited public spaces, such as railway stations and airports, can be found in Google Maps [1].

Indoor positioning is challenging due to the unique properties of the indoor environment. Signal attenuation and reflection limit the applicability of traditional triangulation or time-of-arrival outdoor positioning techniques in indoor environments. Hence, indoor positioning systems are based on various technologies such as Infrared, Ultrasonic, Magnetic Fields, RFID and WLAN technology [2]–[4]. Developers have to make trade-offs between accuracy and cost when they choose a technology. In the last decade, hybrid indoor positioning systems have emerged that simultaneously use various sensors and technologies to determine the position so they can enjoy the advantages of technologies applied. For example, systems that use ultrasonic technology can achieve high accuracy, but they also have huge installation costs and may require specific client devices. On the other hand, systems that use WLAN for positioning have low installation costs and the client device can be an arbitrary smart phone, but their accuracy is lower than that of the systems based on ultrasonic technology. A

sufficiently precise, easily accessible and sustainable industrial standard has not been created yet.

Symbolic indoor positioning determines the location as a well-defined part of the building, such as a room or a corridor. Symbolic positions can be considered as a category, thus the symbolic positioning can be converted into a classification problem. Some well-known classifiers accept classes as prediction based on the confidence values. There are some cases when the confidence for each class is relatively small. Hence the accuracy of these classifiers can vary in a moderate range. To boost the performance of these classifiers, a hierarchical grouping of class categories can be introduced.

The creation of the tree structure of the hierarchical grouping can be manual or automatic. The manual creation requires firsthand domain knowledge of the given environment. However, hierarchical clustering algorithms can be applied using the topological description to generate the tree. The topology of the building defines the rooms, their arrangements and their connections in the buildings. Building topology can be modeled with a wide range of tools. Computer-aided Design (CAD) tools and Building Information Model (BIM) are widely used, but their availability is limited.

IndoorGML (Indoor Geographic Markup Language) [5] is a standard defined by the Open Geospatial Consortium (OGC), which is an open format to describe the topology. It stores the data in XML, and describes the building as the topology of rooms. Along the topology, semantic information is also stored as transitions between the rooms.

The major objective of this paper is to investigate the result of the hierarchical clustering algorithm based on the IndoorGML Document.

II. THEORETICAL BACKGROUND

A. Clustering

Clustering is an unsupervised learning method which goal is to discover large groups of objects in the dataset. A group is called a cluster, and the clusters contain similar objects. Cluster elements which belong to the same cluster are more similar while the elements of different clusters are different. This similarity is often measured with some kind of distance function. Objects from different groups are diverse. Clustering methods can be categorized based on their cluster model.

The clustering process can be seen in Figure 1, where the patterns are the objects. Clustering process has three

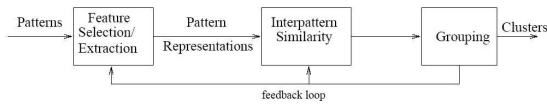


Fig. 1. The steps of clustering process

steps which are feature selection and extraction, interpattern similarity and grouping.

The feature selection process identifies the most effective subset for clustering. The feature extraction makes transformations of available features to create new important features. Pattern or object representation refers to "the number of classes, the number of available patterns, and the number, type, and scale of the features available to the clustering algorithm" [6].

The interpattern similarity is based on the pattern proximity. The pattern proximity is measured by a distance function defined between two patterns. Different distance functions can be defined, but in most cases, the Euclidean distance is used.

The grouping step can be performed in various ways. Hard and fuzzy output of clustering algorithms are distinguished. In hard clustering, each pattern is part of a group exclusively, while in fuzzy clustering, each pattern has a certain degree of membership of each group.

B. Hierarchical clustering

Hierarchical clustering is a method of cluster analysis, which organizes the elements into a tree structure. It is usually a greedy approach, when in each iteration, the local best is selected. The hierarchical clustering is visualized in a dendrogram. Two types of hierarchical clustering can be found, namely top-down and bottom-up.

The top-down type of hierarchical clustering is also called as divisive approach. The top-down approach starts the hierarchical clustering with all the objects in one cluster. Based on different criteria, the cluster is recursively split.

The bottom-up type of hierarchical clustering is also called as agglomerative approach. The bottom-up approach starts the hierarchical clustering with all the objects as an own cluster. Based on different criteria, the cluster is recursively merged.

1) *Linkage method*: The linkage method is used to determine the distance between two clusters. It requires the definition of the used distance metric. There are several commonly used linkage methods [7].

a) *Single linkage*: Single linkage method is also known as Nearest Point Algorithm. It calculates the distance of two clusters, as presented in Equation 1, where i is an object in cluster u and j is an object in cluster v .

$$D(u, v) = \min(d(u[i], v[j])) \quad (1)$$

It is a suitable linkage method in the case of well-separable clusters. Due to the usage of minimal distance, if clusters get too close to one another, it tends to link them and possibly form a split within the cluster. It is sensitive to outliers.

b) *Complete linkage*: Complete linkage method is also known as Farthest Point Algorithm. It calculates the distance of two clusters, as presented in Equation 2, where i is the objects in cluster u and j is the objects in cluster v . It is sensitive to outliers.

$$D(u, v) = \max(d(u[i], v[j])) \quad (2)$$

c) *Average Linkage*: Average linkage method is also known as Unweighted Pair Group Method with Arithmetic Mean (UPGMA). It calculates the distance of two clusters, as presented in Equation 3, where i is the objects in cluster u , j is the objects in cluster v and $|u|$ is the cardinality of cluster u and $|v|$ is the cardinality of cluster v . It is less affected by outliers.

$$D(u, v) = \sum_i \sum_j \frac{d(u[i], v[j])}{|u| * |v|} \quad (3)$$

d) *Weighted Linkage*: Weighted linkage method is also known as Weighted Pair Group Method with Arithmetic Mean (WPGMA). It calculates the distance of two clusters, as presented in Equation 4, where s and t are the two nearest clusters, u is a higher-level cluster combined cluster of s and t and v is the remaining cluster in the forest.

$$D(u, v) = \frac{d(s, v) + d(t, v)}{2} \quad (4)$$

e) *Centroid linkage*: Centroid linkage is also known as Unweighted Pair Group Method with Centroid (UPGMC). In a cluster of points, the centroid is the point that has the average coordinates of all the objects of the cluster. Instead of calculating the distance based on all the cluster objects, only the centroid point is used. Equation 5 shows the distance calculation of cluster s and cluster t , where c_s is the centroid of cluster s and c_t is the centroid of cluster t .

$$D(s, t) = ||c_s - c_t|| \quad (5)$$

The new centroid point is calculated using all the objects in the new cluster.

f) *Median linkage*: Median linkage is similar to the centroid linkage and it is also known as Weighted Pair Group Method with Centroid (WPGMC). It calculates the distance by the same formula as seen in Equation 5.

When s and t clusters are merged into a new cluster u , the centroid of the new cluster is the average of the centroid of cluster s and the centroid of cluster t .

g) *Ward linkage*: The ward linkage method aims to minimize the total within-cluster variance. It is also known as incremental algorithm. Equation 6 shows the formula of the distance calculation, where cluster s and cluster t are merged into a cluster u , v is an unused cluster in the forest and $T = |v| + |s| + |t|$ and $|x|$ is the cardinality of any cluster.

$$d(u, v) = \sqrt{\frac{|v| + |s|}{T} d(v, s)^2 + \frac{|v| + |t|}{T} d(v, t)^2 - \frac{|v|}{T} d(s, t)^2} \quad (6)$$

C. Distance function

Distance functions are used to express the similarity between objects. Similar objects are closer to each other, and their distance approaches zero. Equation (7) shows the general form of distance functions.

$$d : O^2 \rightarrow \mathbb{R}, \quad O : \text{Set of Objects} \quad (7)$$

A distance function is called metric if it fulfills the following criteria:

$$\begin{aligned} i \quad & d(o_1, o_2) \geq 0 \\ ii \quad & d(o_1, o_2) = 0 \leftrightarrow o_1 \equiv o_2 \\ iii \quad & d(o_1, o_2) = d(o_2, o_1) \\ iv \quad & d(o_1, o_2) + d(o_2, o_3) \geq d(o_1, o_3) \end{aligned} \quad (8)$$

III. INDOORGML

The topology of the building can be described using IndoorGML. IndoorGML [5] is a standard defined by the Open Geospatial Consortium (OGC). Each indoor space is considered a closed object specified by virtual or physical boundaries. Figure 2 shows the types of shapes defined with XML Schemas and their hierarchy.

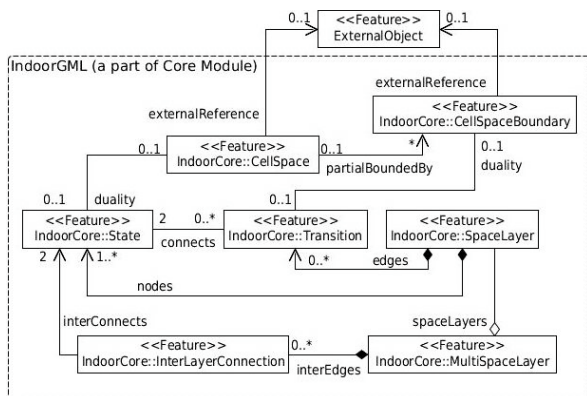


Fig. 2. The structure of IndoorGML

The topology of the environment is stored in created IndoorGML document. The `id` property of `cellSpace` tag is a unique identifier. The name of the room is added to the `metaDataProperty` tag for easier handling and user-friendly consideration. The name of the room is not sufficient for identification due to renaming. In the IndoorGML, two ways are incorporated for the description of an indoor space, both can be used in arbitrary dimensions. The first way uses the vertices of the indoor space. The number of these positions shall be at least four, and the first and the last positions needs to be coincident. The other way is the definition of the bounding box, which can speed up searching processes. It contains the `lowerCorner` and the `upperCorner` tags, which is a pair of positions defining the opposite corners of the indoor space. Hence, the IndoorGML provides the possibility of the usage of coordinate-based distance calculation.

The document contains `Transition` tags, which represents the passage between rooms. `Transition` tag connects two `State` tags, weight and geometry properties can be added for each `Transition`. A `State` tag can be paired to a `CellSpace` tag. Based on the particular building, `SpaceLayer` tags can be utilized for multi-layered space representation. Those represent the floors of the building. The `Transitions` and the `States` can be assigned to a `SpaceLayer`, which stores them in lists. `Transitions` and `States` can be mapped into a weighted, directed graph for way-finding.

IV. EXPERIMENTS

In the experiment, different hierarchical clustering methods with different distance functions are examined. The clustering process is implemented in Python. The IndoorGML document is loaded, and converted into a `DataFrame` using `pandas` package. The grouping of the objects is performed using `SciPy` package, while the tree model is presented using the `tanglegram` package [8].

A. Room representation

The Institute of Information Science (IIS) Building was modeled with IndoorGML [9] standard. The XML files can be created in both automatic and manual ways. The model could be generated from the construction plan, but this plan is not available in the case of this particular building.

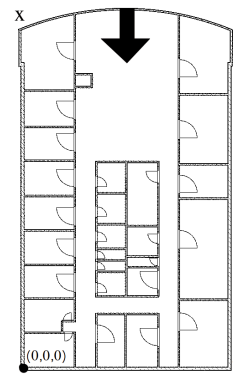


Fig. 3. The structure of the building

The manual creation requires a grid on the building, whose base point is in the bottom left corner of Figure 3. The horizontal axis is the y , and the vertical axis is the x . The z axis is perpendicular to the screen. Currently, the building has a $1m \times 1m$ measured grid in the accessed areas. The coordinates are determined based on the available grid. The identifier of each room is chosen to be the same as in the Miskolc IIS Hybrid Data Set [10]–[12] and the ILONA [13], [14] System with an `uuid` prefix.

The IndoorGML is transformed to be used for clustering purposes.

1) *Feature Selection*: The features are examined based on the practicability for clustering purposes.

Some metadata of the rooms is retained in this step for the evaluation of constructed hierarchies. The physical and virtual boundaries defined using coordinates are selected for presentation. Hence, the lower and upper corner of the bounding box and all of the vertices of the rooms are preserved. Contrary to the layout of the rooms, their transitions are not convenient to create a hierarchy among rooms. However, they can be appropriate for way-finding purposes. In this study, the permeability of the room borders is not incorporated.

2) *Feature Extraction*: After the elimination of unnecessary features, a new feature is introduced. The new feature is the capacity of each rooms. In this experiment, the volumes of the three-dimensional rooms are added. The volume is calculated from the lower and upper corners of the bounding box.

Furthermore, the identifier and the name are merged for indexing purposes. It is required to eliminate these features from the actual grouping process.

B. Similarity

Various distance functions can be defined in a coordinate system. In this study, two distance function is applied.

1) *Euclidean distance function*: The Euclidean distance is the most commonly used distance function. It is a straight-line distance between two points in the Euclidean Space. Euclidean distance is the default option for most of the linkage methods. In addition, some linkage methods only accept this distance function as its metric.

2) *Gravitational force*: The gravitational force-based approach [15] was designed to consider the topology in the classification error calculation. In the calculation process, among the physical distance, a δ function is defined, which represents a virtual distance. This virtual distance can be used to enhance the hierarchical clustering algorithm.

The approach requires the determination of capacity and distance functions. The capacity ($V : Z \rightarrow \mathbb{R}^+$) function maps each room to a positive real value. The distance ($d : Z^2 \rightarrow \mathbb{R}$) function determines how far a room is from another or it measures the dissimilarity of the rooms.

The gravitational force ($F_g : Z^2 \rightarrow \mathbb{R}$) [16] measures the similarity between two rooms. The gravitational force is proportional to the product of their capacity and inversely proportional to the square of their distance. Hence, the gravitational force is non-negative and symmetric derived from the symmetry of distance function and the formulation as seen in (9).

$$F_g(Z_i, Z_j) = \frac{V(Z_i)V(Z_j)}{d(Z_i, Z_j)^2} \quad (9)$$

While the gravitational force represents the similarity between rooms, their difference is required for error calculation. The δ ($\delta : Z^2 \rightarrow \mathbb{R}^+$) function is introduced to represent the difference of two rooms. The δ function is the reciprocal of the F_g function; however, the denominator is increased by

1 in order to avoid division by zero. The δ function can be calculated as seen in (10) and it ranges in $]0, 1[$.

$$\delta(Z_i, Z_j) = \frac{1}{1 + F_g(Z_i, Z_j)} \quad (10)$$

Euclidean distance is used as a distance metric and volume is used as the capacity function for gravitational force-based approach. The δ can be considered as a virtual distance among the rooms.

C. Grouping

The grouping is performed in the bottom-up or agglomerative way. The linkage method and the distance metric is the parameter of the grouping. The result of the grouping is the $(n-1) \times 4$ linkage matrix. The n is the number of the original objects. The first and second columns of the linkage matrix are the ids of the clusters, which will be merged to create a new cluster with an incremented id. The third column is the distance between the two merged clusters. The fourth column is the number of original objects assigned to the newly formed cluster. The linkage matrix is visualised using dendrogram.

D. Comparison

The created cluster models are compared based on the generated dendrograms. A pair of cluster models is represented in a tanglegram. Tanglegram is used to compare tree diagrams. It measures the quality of the two dendrogram alignments as entanglement. For each object, a vector can be established between the two dendrograms. The entanglement is the L norm distance between these vectors. The number of optimization iterations can be specified in order to minimize the entanglement. The entanglement value is the base of the evaluation.

V. RESULTS

During the experiments, the Euclidean distance function and the gravitational force-based distance is examined. The average, centroid, complete, median, single, ward and weighted linkage methods had been tested. In case of Euclidean distance function, all the linkage methods could operate. However, centroid, median and ward linkage methods cannot be applied with gravitational force-based distance or any distance function other than the Euclidean distance.

The dendrogram of the generated tree using complete linkage method with Euclidean distance can be seen in Figure 4. The leaf nodes of the tree are the room identifiers, namely the id and the name. Two main groups of rooms can be distinguished. The height of the dendrogram represents the distance between the clusters.

The dendrogram shown in Figure 4 is an example of the generated results. The trees generated by the same distance function are compared to each other, and then optimized to minimize the entanglement value. These entanglement values are not necessarily symmetric due to the implemented brute force approach.

The entanglement values of the generated tree with Euclidean distance function can be seen in Table I. The values are

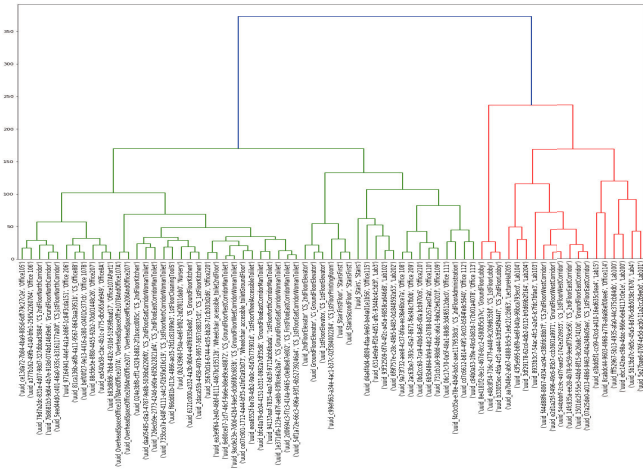


Fig. 4. Dendrogram of Hierarchical Clustering using Complete Linkage Method

TABLE I
ENTANGLEMENT OF METHODS USING EUCLIDEAN DISTANCE OPTIMIZED WITH 10000 ITERATIONS

linkage methods	average	centroid	complete	median	single	ward	weighted
average		2.507	2.7606	1.9155	7.493	3.0704	2.1127
centroid	1.831		3.9437	2.2817	5.662	3.4085	2.0282
complete	3.3803	4.4507		4.0282	7.2676	3.2958	3.7746
median	2.8732	1.8592	3.8873		6.9014	3.3803	2.3099
single	6.1127	6.7887	6.6761	5.2958		5.7183	4.5352
ward	2.5915	3.493	2.3944	2.9859	6.4225		3.2676
weighted	2.338	1.0704	3.2676	6.9014	4.1408		
average value	3.1878	3.3615	3.8216	3.1549	6.7747	3.8357	3.0047

optimized in 10000 iterations. The lowest, 1.0704 entanglement value occurs in the case of centroid and weighted linkage methods. The highest entanglement value in this scenario is single and complete linkage method with value 7.2676. Most of the linkage methods resulted in entanglement value between 3 and 4. However, single linkage has almost two times higher average entanglement value as the other methods.

TABLE II
ENTANGLEMENT OF METHODS USING GRAVITATIONAL FORCE-BASED DISTANCE OPTIMIZED WITH 10000 ITERATIONS

linkage methods	average	complete	single	weighted
average		5.4366	8.9296	4.5634
complete	5.7183		11.9155	5.1549
single	8.1127	12.2254		12.7324
weighted	5.4366	5.2676	11.0704	
average value	6.4225	7.6432	10.6385	7.4836

Table II shows the entanglement values using Gravitational force-based distance. The iteration number is 10000 in this case. The centroid, median and ward linkage methods are not shown in the Table due to the incompatibility of the distance function. The lowest entanglement value is 4.5634 in the case of weighted and average linkage methods. The highest entanglement occurs in the case of weighted and single

linkage methods with value 12.7324. The difference between the average entanglement values of single linkage method to other methods is significant.

TABLE III
ENTANGLEMENT OF METHODS USING GRAVITATIONAL FORCE-BASED DISTANCE OPTIMIZED WITH 100000 ITERATIONS

linkage methods	average	complete	single	weighted
average		4.3099	9.1268	2.9577
complete	4.5915		11.7183	4.5915
single	8.6197	11.5775		10.1127
weighted	3.2676	4.3099	10.3944	
average value	5.4929	6.7324	10.4132	5.8873

Table III shows the entanglement values using Gravitational force-based distance. The iteration number is 100000 in this case. The centroid, median and ward linkage methods are not shown in the Table due to the incompatibility of the distance function. The lowest entanglement value is 2.9577 in the case of weighted and average linkage methods. The highest entanglement occurs in the case of single and complete linkage methods with value 11.7183. The difference between the average entanglement values of single linkage method to other methods is significant.

In Table II and Table III, the increment of the iteration number could decrease the average entanglement value. While in the case of average, complete and weighted methods the average values lessen by at least 0.91, the single method could decrease its averages by 0.22.

The lowest entanglement value in the experiment resulted by centroid and weighted linkage methods using Euclidean distance, and its tanglegram can be seen in Figure 5.

Centroid and Weighted using Euclidean Distance with Optimized Entanglement: 1.0704

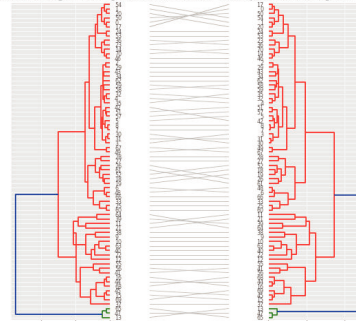


Fig. 5. Centroid and Weighted linkage methods using Euclidean distance with 10000 iterations

The highest entanglement value in the experiment was achieved by using weighted and single linkage methods with gravitational force-based distance. The tanglegram of the two dendrogram can be seen in Figure 6.

The two distance function can be examined by using the same linkage function. The weighted linkage had been selected for the comparison due to the compatibility and lower entanglement values. The tanglegram of the distance functions can be seen in Figure 7 with entanglement value 12.7324.

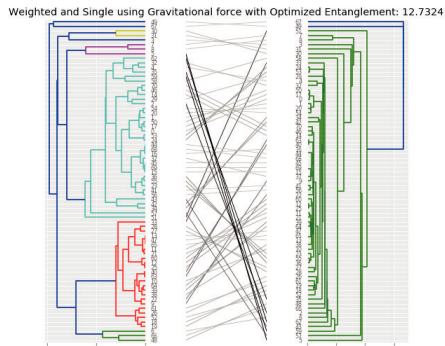


Fig. 6. Weighted and Single linkage methods using Gravitational force-based distance with 10000 iterations

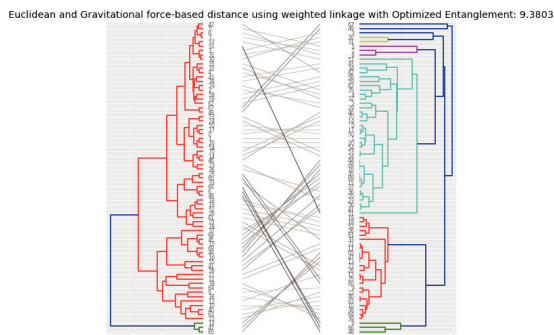


Fig. 7. Euclidean and Gravitational force-based distance using weighted linkage method with 10000 iterations

A. Discussion

Based on the experimental results, two observations can be drawn, one about the linkage functions and one about the distance functions.

The first observation is about the sameness of the linkage functions. Experimental results show, that most of the linkage functions achieved similar average entanglement values. However, the single linkage method shows significant difference in both distance functions. Hence, the single linkage is an outlier method. Based on the Figure 6, the single linkage method does not create explicable hierarchical grouping among the rooms of a building.

The other one is about the usability of the distance functions. Figure 7 shows that the two dendrograms are highly diverse. Euclidean distance can result similar hierarchical clustering using most of the linkage methods. However, these linkage methods seem understandable to the given purpose, the gravitational force-based distance could reflect the topology, the arrangement of the rooms by the distances in the dendrogram.

VI. CONCLUSION

The possibilities of hierarchical clustering for symbolic indoor positioning enhancement purposes are examined in this paper. The physical space is described using IndoorGML.

Euclidean distance function and gravitational force-based distance were used as distance function for the clustering. Average, centroid, complete, median, single, ward and weighted linkage methods had been tested in this experiment. Experimental results show that the single linkage method does not behave like the other linkage methods. However, the usability of the single linkage method can be further examined for this purpose. The gravitational force-based distance could reflect the topology more detailed in the dendrogram.

In the future, the enhancement of symbolic indoor positioning by hierarchical clustering using gravitational force-based distance is planned to be investigated.

ACKNOWLEDGMENT

This research was supported by the grant EFOP-3.6.1-16-2016-00001 (Complex improvement of research capacities and services at Eszterhazy Karoly University).

REFERENCES

- [1] "Google Indoor Maps," <https://www.google.com/intl/en/maps/about/partners/indoormaps/>, [Online; accessed 02-Nov-2015].
- [2] H. Koyuncu and S. H. Yang, "A survey of indoor positioning and object locating systems," *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 5, pp. 121–128, 2010.
- [3] S. Karsai and Z. Tóth, "Comparison of wifi-based indoor positioning techniques," in *Proceedings of the 1st International Conference on Future RFID Technologies*, Nov. 2014.
- [4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [5] J. Lee, K.-J. Li, S. Zlatanova, T. Kolbe, C. Nagel, and T. Becker, "Ogc@indoorgml," *Open Geospatial Consortium standard*, 2014.
- [6] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [7] R. K. Blasfield and M. S. Aldenderfer, "The literature on cluster analysis," *Multivariate Behavioral Research*, vol. 13, no. 3, pp. 271–295, 1978.
- [8] "schlegelp/tanglegram," <https://github.com/schlegelp/tanglegram>, [Online; accessed 13-August-2019].
- [9] K. Ilku and J. Tamas, "Indoorgml modeling: A case study," in *Carpathian Control Conference (ICCC), 2018 19th International*. IEEE, 2018, pp. 633–638.
- [10] "Miskolc IIS hybrid IPS data set," <http://archive.ics.uci.edu/ml/datasets/Miskolc+IIS+Hybrid+IPS>, [Online; Date donated 04-July-2016].
- [11] Z. Toth and J. Tamas, "Miskolc iis hybrid ips: Dataset for hybrid indoor positioning," in *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. Kosice, Slovakia: IEEE, 04.19-04.20 2016, pp. 408–412.
- [12] N. R. T. J. Tóth Zsolt, Magnucz Péter, "Data model for hybrid indoor positioning systems," *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*, vol. 7, pp. 67–80, 2015.
- [13] Z. Toth, "Ilona: indoor localization and navigation system," *Journal of Location Based Services*, vol. 10, no. 4, pp. 285–302, 2016. [Online]. Available: <http://dx.doi.org/10.1080/17489725.2017.1283453>
- [14] D. P. Kun, E. B. Varga, and Z. Toth, "Ontology based navigation model of the ilona system," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, Jan 2017, pp. 000 479–000 484.
- [15] J. Tamas and Z. Toth, "Topology-based classification error calculation for symbolic indoor positioning," in *Carpathian Control Conference (ICCC), 2018 19th International*. IEEE, 2018, pp. 643–648.
- [16] I. Newton, *The Principia: mathematical principles of natural philosophy*. Univ of California Press, 1999.

Significance of implementation of Management Information Systems in the IPO chain for SMEs

Katarína Teplická
 Institute of Earth resources
 Technical university, Faculty BERG
 Košice, Slovakia
 katarina.teplicka@tuke.sk

Abstract— Management information systems support the efficient management of processes, resources, materials, machinery, equipment, people and finance. The strategic goal of creation and management information system is to support the growth of performance and value of business. Software support for SMEs business is very important. The aim of the paper is to point out the importance of implementing MIS in small and medium enterprises in the IPO chain, to examine the selection of optimal software for MIS based on preferred criteria. We selected specific software products for SMEs in Slovakia, that are offer to IT market. We analyzed risks and benefits in area of implementation MIS in SMEs in Slovakia. The results of MIS implementation point to the benefits generate quality information for decision-making, saving time downtime, reducing administrative staff, increasing inventory turnover, slimming down the count of IT departments.

Index Terms— software products, management information system, controlling, dashboard,

I. INTRODUCTION

Improving business success means improving business processes and using process-oriented support tools to achieve key factors such as efficiency, economy and productivity. The use of business information software system support is very effective and necessary in the long run for SMEs.

Business processes cannot be managed without changes and improvements, which can ultimately affect the price of products, demand, meet customer requirements, increased revenue from product sales, to decrease total costs, and to create profit growth. In addition to financial benefits, it is also necessary to perceive another aspect of this approach, namely to improve the information transfer system at individual management levels, to provide feedback in synergy with the customer, to eliminate ineffective activities in the chain of business processes [4].

The business transformation process includes all business activities (investment activity, supply activity, personnel activity, production activity, sales activity, economic activity and management activity), by means of which it ensures fulfillment of set tasks and goals. Transformation process change input to output in IPO chain [1]. The fundamental role of businesses is to optimize the relationship between inputs and

outputs. All activities in the IPO chain are monitored through software tools, applications that form the basis for their management. In SMEs it is important to implement a comprehensive information system that provides all the information for effective management (fig.1).

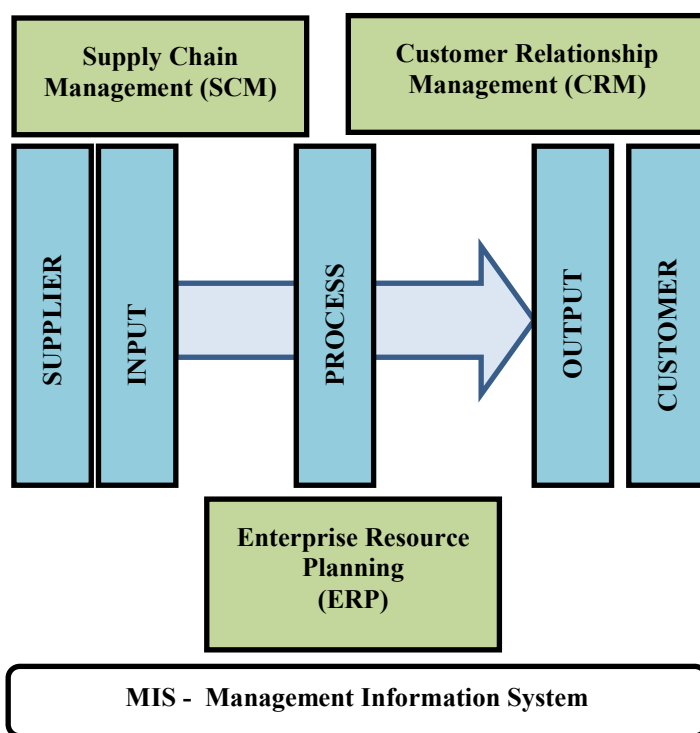


Fig. 1. Management information System in IPO chain.
 Source: own source

Management information system consists of ERP, CRM and SCM systems.

Pervan, Dropulic (2019) present that creation of management information system in SMEs is very important and the implementation of accounting systems to MIS effects performance of SMEs. The quality of accounting system implementation to MIS significantly and positively influenced changes of internal reporting in Croatians firms. Results of research in Croatian firms show that MIS sellers should put special focus on implementation of business analytics modules [2].

Jayawickrama (2019) present factors that they influence structure of MIS in UK SMEs. Knowledge retention is vital for

various modules implementations in small and medium-sized enterprises (SMEs). The factors that influence structure of MIS are documentation, human capital, project management and organizational culture. In addition to these factors, there are barriers in SMEs that do not allow the implementation of management information systems. The main reason is the financial sources and operating staff of MIS [5].

Čarnický (2003) in his contribution presents main problems of implementation of EIS in the SMEs. On the basis of analysis results he refers to the fact that managers in the enterprises do not apply adequately EIS in the area of corporate management. The biggest problem appears to be insufficient information knowledge and training of top and other managers for application of these efficient in-formation systems on the managerial levels of company management [7].

Lajsic.H.(2019) presents that there is no efficient strategic management of human resources without the support of information technologies and information systems. The development of information technologies and information systems dramatically changes the management of human resources. The strategic readiness of information capital (IT and systems) is the most meaningful measure of the value of the information capital of the organization. The greatest availability of information capital in human resources management is measured by the level of development of online transaction processing systems, whether it is a module of an ERP system, or that it is HRIS (Human Resources Information Systems). Another technology that has even more strategic significance is BI (Business Intelligence) technology [6]. Based on the statements of several authors, we can state that information and its processing are important for the management of all processes in the IPO chain. The importance of introducing MIS in SMEs is increasing and plays a strategic role in business (fig.2).

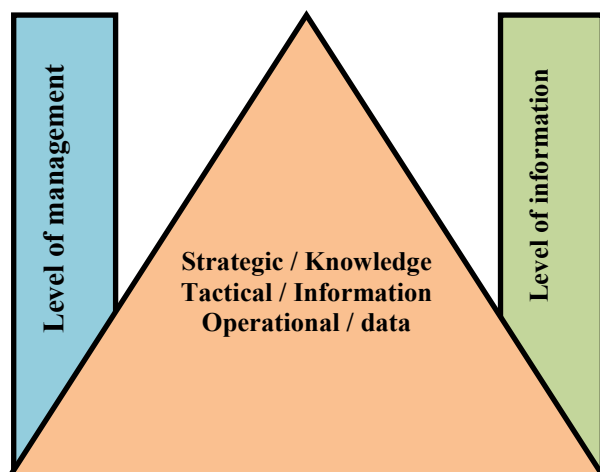


Fig. 2. Relationship between information and management levels.
Source: own source

Management information system consists of ERP, CRM and SCM systems. Through ERP systems key business processes can be successfully managed and enable the enterprise as a whole to build a effective concept of management information system. The essence of ERP systems

is to achieve transparency in business activities, processes, planning, calculating, budgeting, process decisions, orders, performance, customers, and products [3]. All businesses are looking for an optimal solution for implementing ERP systems or MIS systems. Currently, there are a number of companies offering management information systems - SAP, MICROSOFT SLOVAKIA, ORACLE, SOFTIP, SIEMENS, NESS and other [4]. ERP is an effective tool for business process planning and management. The ERP system integrates all business processes in the firm. Benefits of ERP systems are: automation and integration of main processes, data sharing, standardization, real-time information creation and availability, historical data processing, complex approach to ERP solutions.

CRM is a business approach that enables you to take customer relationships to a new level. His primary task is to establish good long-term relationships with clients.

Saydasheva and Sharafutdinova (2018) present CRM system as determines an original technique, according to which it is possible to distinguish a typical customer's "portrait" and his preferences. CRM system is used too marketing activity management process in the firms. CRM system consists of modules for organizations of service. His task is to form an idea about of customers, kind of promotion, focus services to customers [9].

Supply chain management (SCM) is the broad range of activities required to plan, control and execute a product's flow, from acquiring raw materials and production through distribution to the final customer, in the most streamlined and cost-effective way possible.

Sarkar, Chung presented (2019) that for a complex product production, any flexible manufacturing system with a work-in-process inventory is recommended for a supply chain management (SCM) system. Building a flexible manufacturing system increases the total cost of the supply chain. The aim of the supply chain is to reduce the total cost when demand during the lead time [10].

The interconnection of all three ERP, CRM, and SCM systems is the basis for building a management information system for SMEs.

II. RESEARCH OF USING OF MIS SYSTEMS IN SMEs IN SLOVAKIA

MIS systems represent the integration of key processes in the IPO chain in enterprises. MIS systems are aimed at improving and streamlining business processes from entry to exit, which is the IPO chain. The research was carried out in small and medium-sized enterprises in Slovakia in the number of 256 addressed enterprises. The questionnaire was returned by 248 companies, which we consider a 97% return on the questionnaires, which is very important for the research in terms of information reliability.

The research of enterprises was focused only on industrial enterprises in SMEs. The established hypotheses were oriented on MIS and their implementation in SMEs.

H1: The reason for introducing MIS is to improve performance in the firms.

H2: The most commonly used information system modules are finance.

H3: The cost of implementing the MIS is high for SMEs and beyond the funding plan.

Results of research for established hypotheses were oriented on MIS and their implementation in SMEs. Results of first hypothesis are presented in fig. 3

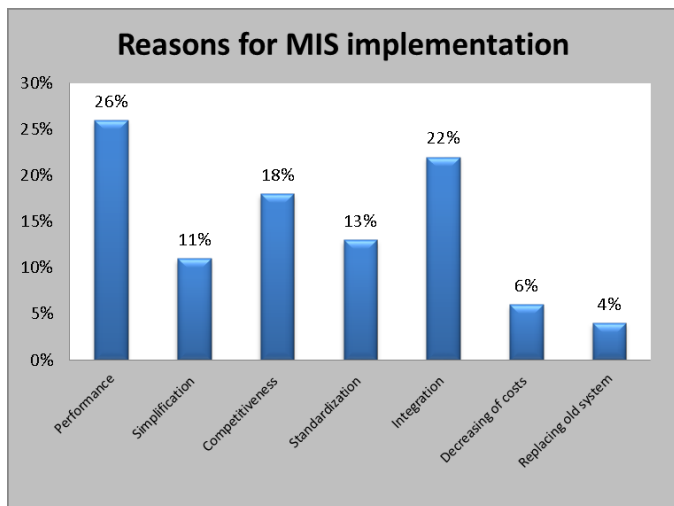


Fig. 3. Reasons of MIS implementation in Slovakia.[8]

H1: The reason for introducing MIS is to improve performance in the firms.

The hypothesis has been confirmed because companies really implement MIS to improve business performance.

MIS systems are made up of various modules that are tied to business processes. Based on the IPO chain, businesses decide what modules they need for their business processes. Most often, businesses use modules that are bound to enterprise inputs and outputs. Basic software modules are logistics, human resources, production, finance, and sale. The proportion of implemented modules in MIS systems is presented in fig.4.

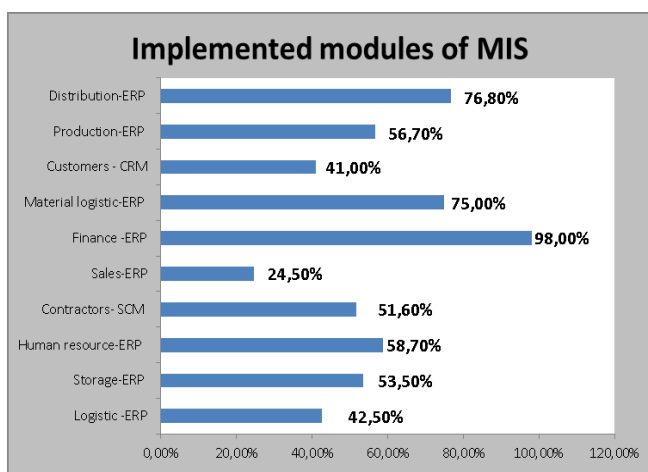


Fig. 4. Implemented modules of ERP in the firms in Slovakia.[8]

H2: The most commonly used information system modules are finance.

The hypothesis has been confirmed because companies really implement module of MIS for finance 98%.

The average age of using MIS systems is getting shorter. In small businesses, the age of MIS systems is reduced from 7 to 5 years, in large and medium-sized enterprises the age of MIS systems is reduced from 8 to 6 years. There are no major barriers to implementing MIS systems in large enterprises, and there is no need to make major changes to information systems.

In small businesses, the implementation of MIS systems is challenging because of major constraints, and extensive changes to information systems and sub-adjustments are needed. Continuous innovation of MIS systems is important in terms of functionality, as process changes, market conditions, and legislative conditions are constantly occurring. The second factor that is crucial for MIS systems innovation is outdated technology and inflexible architecture. MIS systems are also being introduced into businesses in terms of size. The results of MIS systems implementation in enterprises depend mainly on financial means and this represents the main problem in the implementation of MIS systems in Slovak enterprises in SMEs (fig.5).

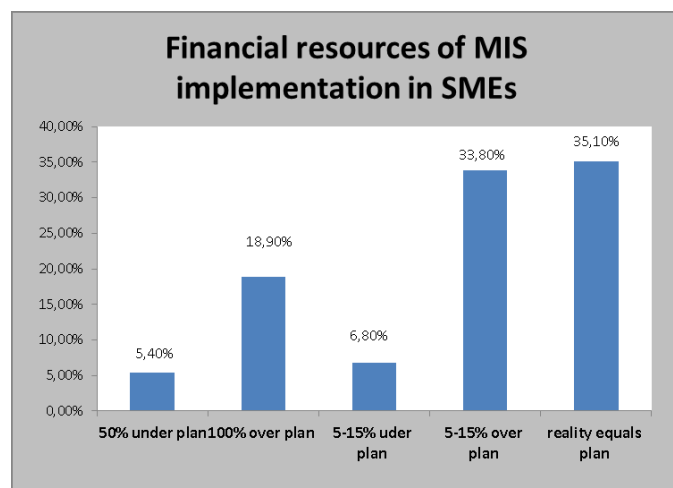


Fig. 5. Difference between actual costs and plan.[8]

H3: The cost of implementing the MIS is high for SMEs and beyond the funding plan.

The hypothesis has been confirmed because costs for implementation of MIS are high and create 5-15% of financial sources over plan.

III. ANALYSIS OF RISKS AND BENEFITS OF MIS IN SLOVAK COMPANIES

Creating of MIS systems in enterprises must be based on IPO chain and business processes. The management information system enables enterprise management to access various statistics that are important for decision making. The MIS system must be flexible, regular, up-to-date, fast, and available. The strategic goal of the MIS system is to support

business performance and value growth. Quality MIS systems are beneficial to businesses and mean an effective management tool.

The introduction of MIS systems in enterprises presents risks and benefits that bring opportunity. We present risks and opportunities in SMEs on base of realized research. Implementation of MIS brings not only financial risks. Implementation is a complex process that involves risks to the software vendor and the company that deploys the MIS system. The size of the risk depends on how the MIS system implementation project is managed. The risks created by implementing MIS system in SMEs are presented in tab. 1.

TABLE I. RISKS OF MIS IMPLEMENTATION IN SLOVAK COMPANIES

AREA	RISKS
Management	Problems in changing mentality and worker habits Leaders' orientation towards short-term Goals, which is not rational in terms of strategic goals.
Supplier	Bad reputation of the supplier Financial losses Weakening market position Poor business ability
Organization structure	Problems associated with changing the organizational structure of the business. Management decentralization issues related to increased decision-making autonomy. Limited time to adapt verified practices and management concepts to specific business conditions.
Information	Bad choice of MIS system Unstable database Unreliable results
Finance	Lack of funds to buy the necessary computing. Few opportunities for adequate financial incentives.
Implementation of ERP system	Unproductive methodology Delay of module delivery dates Budgets exceeded MIS system failures Incorrect MIS criteria

Source: [8]

The core of these MIS systems is system integration. Creating a quality information system for businesses minimize risks and maximize opportunities. The goal of system integration is to create and maintain an integrated information system that uses the potential of available information and communication technologies. The management information system in a small and medium-sized enterprise must capture all business processes with software support. For small and

medium-sized enterprises, it is preferable to implement an integrated management information system than partial software applications for individual processes. In addition to technical and economic advantages, it also presents other advantages that are presented in tab.2.

TABLE II. BENEFITS OF MIS SYSTEM IMPLEMENTATION IN SLOVAK COMPANIES

Area	Benefits
Planning, Motivation	Improvement of the structure of plans, extension of plans containing external parameters, assets and liabilities, financial flows and selected indicator values. Improve the accuracy and quality of planning by identifying them by senior staff of those departments that require increased attention. Improving the effectiveness of the enterprise's management activity by deepening the knowledge of the weaknesses and strengths of the business, its bottlenecks. Increasing the motivation of executives at lower levels of management by linking tasks to the motivation system of their assessment.
Efficiency	Improving control activity. Possibility of monitoring, measurement of indicators. Higher quality of processed analyzes.
Strategic Goal	Greater credibility of the data. Precise formulation of tasks, business objectives. Decentralization of decision making. Improving the flow of information. Improved knowledge of middle-level managers.

Source: [8]

IV. ANALYSIS OF SOFTWARE PRODUCTS ON SLOVAK MARKET

Companies developing MIS systems have a lot of competition in the market. Companies track customer requirements, changes in technology and processes, and customize their products to meet customer requirements. Major suppliers include SAP, Oracle, Microsoft (tab.3). A large number of companies in Slovakia are involved in the development of information systems and their implementation in companies. The opportunities of these information companies depend on customer requirements in the area of management information systems. According to the results of the Center for Research of Information Systems CVIS in Slovakia, domestic MIS systems and domestic manufacturers of MIS systems dominate. The main reason is the knowledge

of Slovak market conditions and customers. These manufacturers can quickly respond to changes in MIS systems and adapt systems to customer requirements. The research included finding software products for companies in SMEs. We chose Slovak manufacturers and also software vendors - SAP, ORACLE, MICROSOFT, who are represented on the Slovak IT technology market. Each company has in its portfolio three basic systems (ERP, SCM, CRM) for building a management information system (tab.3).

TABLE III. PRODUCTS OF MIS SYSTEM IN SLOVAK COMPANIES

COMPANIES	PRODUCTS
SAP AG SMEs	SAP Business One SAP Business By Design SAP ERP SAP SUPPLY CHAIN MANAGEMENT SAP SUPPLIER RELATIONSHIP MANAGEMENT SAP ERP FINANCIALS SAP ERP HUMAN CAPITAL MANAGEMENT SAP CRM SAP SERVICE DELIVERY
ORACLE SMEs	Oracle eBusiness Suite Customer Relationship Management Financial Management Human Capital Management Manufacturing Order Management Project Management Supply chain Management Supply chain Planning Quality Management Plan Managers Dashboard
MICROSOFT SMEs	Microsoft Dynamics CRM Microsoft Dynamics ERP Microsoft Dynamics GP Microsoft Dynamics NAV Microsoft Dynamics SCM Microsoft Dynamics RMS

Source: [8]

V. ANALYSIS OF MIS IMPLEMENTATION IN SELECTED COMPANY – CASE STUDY

A high-quality MIS system allows the company to generate outputs that can be used by all stakeholders - employees, customers, suppliers, potential competitors. These outputs are an effective tool for managing business processes. Effective management in SMEs is important instrument for competitive advantages. Slovak companies introduce MIS very carefully and slowly.

We analyzed all processes in IPO chain in selected company and software support of those processes. Then we analyzed economical aspects of this software support. Within the IPO chain, various software products (fig.6) are used in selected company to manage the purchase (TANGRAM), supply (TANGRAM), storage (SOFTIP),

production (MONET ABC/ABB), sales (TANGRAM), distribution (TANGRAM), finance and accounting (OMEGA MAXI). Total output from each process can be done using statistical reports by software ONIX. All processes are monitoring by software support in the company and total amount of this software is 23 261 euro (tab.4). Software was buying step by step.

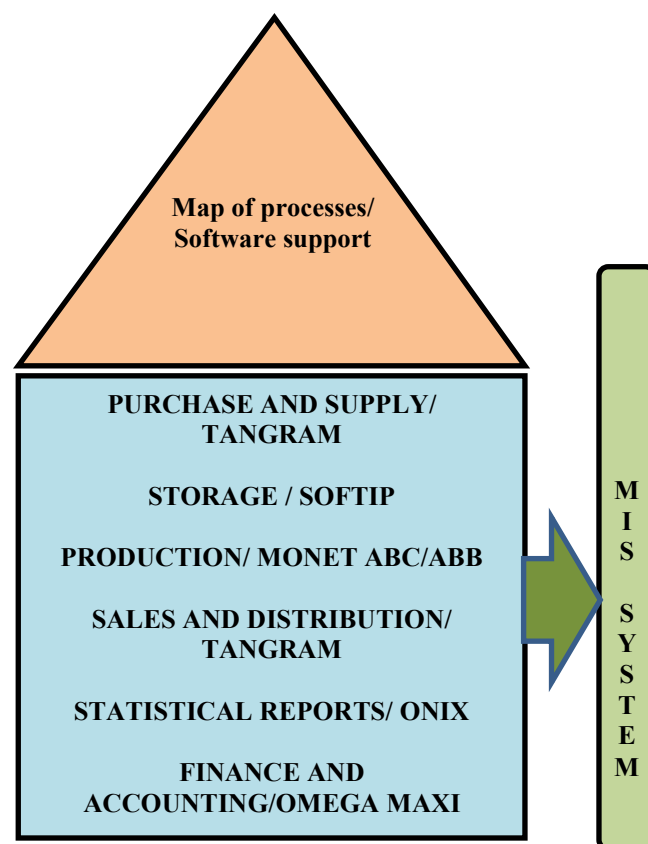


Fig. 6. Software support for processes in company
Source: Internal information of selected company .[8]

All software applications for individual business processes can be merged into a single package and this is the MIS system. An important criterion for MIS systems is the ability to solve problems in individual business processes through data collection. MIS system manufacturers try to target a wide range of customers and create all modules in the system to cover every business process. Software manufacturers also focus on specific areas such as banking, insurance, education, health, public administration.

MIS systems are created in Slovakia but some MIS systems are also purchased abroad. A high share of MIS systems is in Denmark, the Czech Republic, Germany and the United Kingdom. Key business processes that companies prefer as the basis of MIS systems are human resources, economics, logistics, and manufacturing. The advantage of the MIS system is the interconnection of all partial applications in the company and the creation of a quality information system. Partial outputs from individual processes are grouped into one database, which allows monitoring the dependencies and connections of individual processes. This approach is an essential advantage

of MIS systems. The research also included the search for the optimal software for the selected company for the implementation of a comprehensive management information system based on the preferred technical, technological and economic requirements of the company (tab.4).

TABLE IV. ECONOMIC ANALYSIS OF SOFTWARE IN SELECTED COMPANY

PROCESS	SOFTWARE	PRICE
Purchase and supply	TANGRAM	793 €
Storage	SOFTIP	398 €
Production	MONET ABC	1980 €
Production	MONET ABB	3290 €
Sales and distribution	TANGRAM	1600 €
Statistical reports	ONIX	1600 €
Finance and accounting	OMEGA MAXI	1366 €
SUMMARY		23 261 €

PROCESS	MIS SOFTWARE	PRICE
Purchase and supply	SOFTIP	14 400 €
Storage	PROFIT PLUS	
Production	GRADIENT 5	11 107 €
Production	ERP KARAT	15 650 €
Sales and distribution	SOFTIP MIS	14 630 €
Statistical reports	ONIX	13 560 €
Finance and accounting	SAP HYBRIS	16 750 €

Source: Internal information of companies in Slovakia .[8]

CONCLUSION

The successful use of MIS systems in the practice of industrial enterprises means increasing performance and gaining competitive advantage in the market. The implementation of MIS systems takes longer and brings positive benefits to business process management companies. The MIS system is the interconnection of management and information flow in businesses that need to be processed for controlling and preparing dashboards that visually track business results. The main reason for implementing MIS in enterprises was the level of performance. Companies have implemented various modules within the MIS system, most often finances, supplies, supplies, human resources. In terms of financial coverage of the implementation of MIS systems, the funds were overdrawn over the plan. The greatest risks identified in the implementation of MIS systems were business, supplier, project management, MIS system selection, and general risks. Significant companies developing MIS systems include SAP, ORACLE, and MICROSOFT in Slovakia. In the selected company, we proposed to replace the individual software products for business processes with a management information system. We presented MIS prices so that the company could choose the optimal solution. For small and

medium-sized enterprises, the underlying problem is the financial aspect of implementing MIS systems. Research has found that businesses pay more when buying sub-modules than they would have paid when buying a complex MIS. At the same time, the incompatibility of individual software applications is a significant drawback. The research concludes that it is strategically better for SMEs to buy an integrated MIS system despite high prices. Every company has to plan the funds so that it can provide a quality and efficient information system without future risks. The possibilities of purchasing MIS systems in Slovakia are wide and prices of MIS systems range from 10 000 EUR - 100 000 EUR. Slovak information companies also offer service of MIS systems and the creation of new modules that are compatible with the basic system. IT companies offer in their MIS system a perfect overview of the company in one system, regulation in business processes, view about customer relationship and a pillar of performance growth.

ACKNOWLEDGMENT

This contribution was supported by project KEGA 002 TUKE -4/2017 Innovative didactic methods in education at university and KEGA 049TUKE-4/2019 Education of foreign students in Slovakia.

References

- [1] Š. Kassay, "Firm of international class," Nové Zámky, Strateg, 2001.
- [2] I. Pervan, I. Dropulic, "The impact of integrated information systems on management accounting: Case of Croatia," in Journal of Contemporary Management, vol.24, issue 1, p.21-38, 2019.
- [3] M. Potkány, A. Šatanová, "Controlling – modern tool of company control," in Economic journal, Bratislava, 2004, p. 148-165.
- [4] T. Petřík, "Economical and financial management of the firm," Praha, Grada Publishing, 2005.
- [5] U. Jayawickrama, SF. Liu, MH. Smith, P. Akhtar, M. Al Bashir, "Knowledge retention in ERP implementations: the context of UKSMEs," in Production planning & control, vol.30, issue 10-12, p. 1032-1047, 2019.
- [6] H.Lajsic, "Strategic human resources management and business intelligence systems in the purpose of social and economic development," in Casopis za ekonomiju I trzisne komunikacije, vol.9, issue 1, p. 53-68, 2019.
- [7] Š. Čarnický, "New position and tasks of management information systems of enterprises," in Economic journal, vol. 51, issue 5, p.570-591, 2003.
- [8] M. Demečko, "Eliminating production process losses for productivity gains," Doctoral thesis, Faculty of Engineering, Kosice, 2019.
- [9] VA. Saydasheva, NS. Sharafutdinova, "Marketing activity management through the effective CRM system," in IIOAB journal, vol. 9, p. 17-21, 2018.
- [10] M. Sarkar, BD. Chung, "Flexible work in process production system in supply chain management under quality improvement," in International journal of Production research, vol. 57, issue 14, p. 30-41, 2019.

Queueing Models of Systems with Non-Homogeneous Customers and their Applications in Computer Science

Oleg Tikhonenko
Institute of Computer Science
Cardinal Stefan Wyszyński University in Warsaw
 Warsaw, Poland
 o.tikhonenko@uksw.edu.pl

Marcin Ziółkowski
Institute of Information Technology
Warsaw University of Life Sciences – SGGW
 Warsaw, Poland
 marcin_ziolkowski@sggw.pl

Abstract—In the paper, we present a comparatively new direction of queueing theory named the theory of queueing systems with non-homogeneous (or random volume) customers. We analyze basic principles, performance parameters of the theory and discuss its problems. Some simple examples and results for models of systems with random volume customers are attached as well.

Keywords—random volume customer, buffer space capacity, total customers' volume, loss probability, probability of unit volume losing, Laplace-Stieltjes transform

I. INTRODUCTION

Queueing systems and networks are widely used as mathematical models of many real objects, such as markets, communication and computer networks, etc. Classical queueing models are often used for determination of communication networks' nodes characteristics. One of the most important of them is probability of message's losing that depends on demands' arrival process characteristics, buffer space capacity of the node and distributions of message's volume and its service time.

In late 1970s, it was noticed by engineers from different countries (see e.g. [1–4]) that analytical methods of classical queueing theory are generally insufficient for solving the problem of communication nodes buffer space capacity determination. Indeed, let $\eta(t)$ be a number of customers present in a queueing system at some time instant t . It is known that almost all characteristics of the system can be obtained via analysis of the process $\eta(t)$. But different messages (customers) in a network consist, as a rule, of different numbers of bytes, in other words, they have different volumes.

Then, one of the most important process characterizing a queueing system with customers of random volume becomes the total volume of customers present in it at an arbitrary time instant. We also have to take into account the possible dependence between volume of the customer and his service time. The value of total customers' volume is usually limited by some deterministic value, known as buffer space capacity, and its distribution depends on the joint distribution of the customer volume and his service time.

In the paper, we give the basic principles of a new theory called the theory of queueing systems with non-homogeneous customers. We also characterize some of its problems and present obtained results.

The paper is organized as follows. Part II contains basic notions of the theory and presents some its peculiarities in

comparison with classical queueing theory. Part III contains classification of models of the theory and two simple examples of these models. In Parts IV and V we present loss characteristics and discuss some specific problems of their determination. In Part VI we present an analog of Little's formula for systems with random volume customers and show necessity of taking into account the dependence between customer's volume and his service time. Part VII presents conclusions and final remarks.

II. THE BASIC NOTIONS OF THE THEORY

First, we have to assume that each customer in the system under consideration is characterized by some random volume ζ , where ζ is a non-negative random variable (RV). Let $\sigma(t)$ be the sum of volumes of all customers present in the system at time instant t . The process $\sigma(t)$ is called total volume (of customers in the system at time instant t). This process is a generalization of $\eta(t)$ one. Indeed, we have $\sigma(t) = \eta(t)$ numerically, if we assume that $\zeta \equiv 1$. We also assume that volumes of different customers are independent and do not depend on the customers' arriving epochs. Note that we can generalize our model by supposition that each customer is characterized by random l -dimensional vector of indications, but, in this paper for simplicity, we limit oneself to the case of $l = 1$.

Remind that, in classical queueing theory (see e.g. [5]), one usually assumed that service time distribution is a characteristic of a server, i.e., in non-priority queues, all customers have identically distributed service times. It is clear that, in the case of random volume customers, one makes sense to take into account a possibility of dependence between customer's volume ζ and his service time ξ . This dependence can be defined by the following joint distribution function (DF):

$$F(x, t) = P\{\zeta < x, \xi < t\}. \quad (1)$$

Then, we have for DFs of RVs ζ and ξ respectively:

$$L(x) = P\{\zeta < x\} = \lim_{t \rightarrow \infty} F(x, t) = F(x, \infty),$$

$$B(t) = P\{\xi < t\} = \lim_{x \rightarrow \infty} F(x, t) = F(\infty, t).$$

In a real system, the total volume can be limited by constant value $V > 0$ that is called buffer space capacity of the system. Then, inequality $0 \leq \sigma(t) \leq V$ holds. We assume that buffer space is occupied by a customer at the epoch he arrives

and is released entirely at the epoch he completes service. If the value V is finite, it leads to additional losses of customers. A customer of volume x , which arrives at epoch τ when there are idle servers or waiting positions, will be admitted to the system, if $\sigma(\tau-0) + x \leq V$. Otherwise (if $\sigma(\tau-0) + x > V$), the customer will be lost. In the first case, we have

$$\eta(\tau) = \eta(\tau-0) + 1, \quad \sigma(\tau) = \sigma(\tau-0) + x,$$

in the second one, we obtain:

$$\eta(\tau) = \eta(\tau-0), \quad \sigma(\tau) = \sigma(\tau-0).$$

If t is an epoch of service termination of a customer having volume x , we obtain

$$\eta(t) = \eta(t-0) - 1, \quad \sigma(t) = \sigma(t-0) - x.$$

III. CLASSIFICATION OF MODELS AND SIMPLE EXAMPLES

Consider firstly a classical queueing system of sufficiently general type, e.g. $GI/G/n/m$, $1 \leq n \leq \infty$, $0 \leq m \leq \infty$. The structure of this system is shown schematically in Fig. 1. It is clear that, within limits of this model, we cannot strictly determine total volume characteristics or loss ones causing by buffer space limitation.

A system $GI/G/n(m, V)$ with random volume customers and limited (by V) buffer space is shown schematically in Fig. 2.

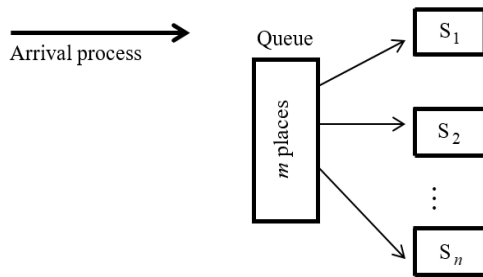


Fig. 1. Classical queueing system

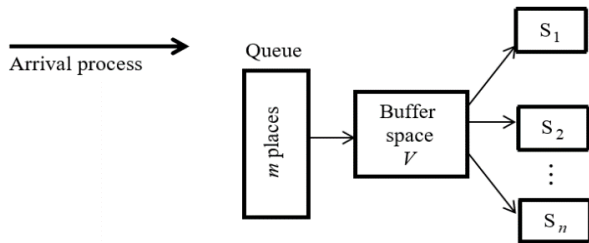


Fig. 2. Queueing system with random volume customers

Obviously, the system shown in Fig. 2 generally differs from classical one. As an example, compare mechanisms of customers' losing in both systems under assumption that there are no other limitations except those shown in Fig. 1 and Fig. 2.

In both systems, a customer is lost, if, at the epoch τ of his arrival, there are $n+m$ other customers in each of them ($\eta(\tau-0) = n+m$). In the system $GI/G/n(m, V)$, a customer having volume x is lost not only if $\eta(\tau-0) = n+m$, but also if $\sigma(\tau-0) + x > V$. It means that an admission or losing of a customer partly or completely depends on his volume and total volume of other customers present in the system.

Classification of models of systems with random volume customers is determined by the type of RVs ζ and ξ dependence and the value V . Then, we can distinguish four classes of these models.

- 1) Models with $V = \infty$ and independent RVs ζ and ξ ($F(x, t) = L(x)B(t)$).
- 2) Models with $V < \infty$ and independent RVs ζ and ξ .
- 3) Models with $V = \infty$ and dependent RVs ζ and ξ ($F(x, t) \neq L(x)B(t)$).
- 4) Models with $V < \infty$ and dependent RVs ζ and ξ .

Note that an analysis of models of the first class can be carried out by tools of classical queueing theory, since, in this case, a distribution of total volume $\sigma(t)$ is completely determined by distribution of number of customers present in the system $\eta(t)$ and DF $L(x)$ of customer's volume.

The main characteristics of models from classes 2 and 4 are the distribution of number of customers present in the system at arbitrary time instant or in steady state and some loss characteristics (e.g. steady-state loss probability).

The main characteristic of models from class 3 is the distribution of total volume at arbitrary time instant or in steady state. Various models of queueing systems with random volume customers can be found in [6–9].

Consider now two simple examples of the models belonging to classes 2 and 3.

A. System $M/M/1(\infty, V)$ with customer's service time independent of his volume

This model is a simple special case of more general queueing system $M/M/n(m, V)$ (see e.g. [6, 9]). Consider firstly for comparison a classical Markov queueing system $M/M/1/\infty$. Denote by a parameter of customers' arrival process and by μ parameter of service time ξ distribution.

Let $P_k(t) = P\{\eta(t) = k\}$, $k = 0, 1, \dots$, be probability of presence of k customers in the system at time moment t . It is known [9] that functions $P_k(t)$ satisfy the following differential equations:

$$\begin{aligned} P'_0(t) &= -aP_0(t) + \mu P_1(t); \\ P'_k(t) &= aP_{k-1}(t) - (a + \mu)P_k(t) + \mu P_{k+1}(t), \quad k = 1, 2, \dots \end{aligned} \quad (2)$$

To analyze the system $M/M/1(\infty, V)$ (this system is not Markov), we need to introduce the following functions:

$$G_k(x, t) = P\{\eta(t) = k, \sigma(t) < x\}, \quad k = 1, 2, \dots$$

Then, we have for this system that $P_k(t) = G_k(V, t)$, $k = 1, 2, \dots$. Let $L(x)$ be DF of customer's volume ζ . As RVs ζ and ξ are independent in the system $M/M/1(\infty, V)$, we obtain the following equations for functions $P_k(t)$ of this system, instead of equations (2):

$$\begin{aligned} P'_0(t) &= -aP_0(t)L(V) + \mu P_1(t); \\ P'_1(t) &= aP_0(t)L(V) - a \int_0^V G_1(V-y, t) dL(y) - \mu P_1(t) + \mu P_2(t); \\ P'_k(t) &= a \int_0^V G_{k-1}(V-y, t) dL(y) - \\ &\quad - a \int_0^V G_k(V-y, t) dL(y) - \mu P_k(t) + \mu P_{k+1}(t), \\ &\quad k = 2, 3, \dots \end{aligned} \quad (3)$$

In steady state that exists when $\rho = a/\mu < \infty$, the following finite limits exist

$$p_0 = \lim_{t \rightarrow \infty} P_0(t);$$

$$g_k(x) = \lim_{t \rightarrow \infty} G_k(x, t), \quad p_k = \lim_{t \rightarrow \infty} P_k(t) = g_k(V), \quad k = 1, 2, \dots$$

From the system of differential equations (3), we obtain the steady-state algebraic ones:

$$0 = -ap_0L(V) + \mu p_1;$$

$$0 = ap_0L(V) - a \int_0^V g_1(V-y) dL(y) - \mu p_1 + \mu p_2;$$

$$0 = a \int_0^V g_{k-1}(V-y) dL(y) - a \int_0^V g_k(V-y) dL(y) - \mu p_k + \mu p_{k+1}, \quad k = 2, 3, \dots$$

Solution of these equations takes the form:

$$g_k(x) = p_0 \rho^k L_*^{(k)}(x), \quad k = 1, 2, \dots,$$

where $L_*^{(k)}(x)$ is k -fold Stieltjes convolution of DF $L(x)$, or, in other words, probability that the sum of k independent identical distributed RVs of the same distribution (with DF $L(x)$) is less than x . Then, for probabilities p_k , we obtain:

$$p_k = p_0 \rho^k L_*^{(k)}(V), \quad k = 1, 2, \dots$$

From the normalization condition $\sum_{k=0}^{\infty} p_k = 1$, we have:

$$p_0 = \left(1 + \sum_{k=1}^{\infty} \rho^k L_*^{(k)}(V) \right)^{-1}.$$

Now, we can easily obtain a relation for customer's loss probability P_{loss} using the following equilibrium equation:

$$a(1 - P_{loss}) = \mu \sum_{k=1}^{\infty} p_k = \mu(1 - p_0)$$

that follows from the following evident fact: in steady state, mean number of customers accept to the system during a unit of time equals to mean number of customers complete their service during this time, whereas $P_{loss} = 1 - (1 - p_0)/\rho$.

E.g. if customer's volume has an exponential distribution with parameter f ($f > 0$), we obtain:

$$p_0 = \begin{cases} \frac{1-\rho}{1-\rho e^{-(1-\rho)fV}}, & \text{if } \rho \neq 1, \\ (1+fV)^{-1}, & \text{if } \rho = 1; \end{cases} \quad (4)$$

$$p_k = p_0 \rho^k \left[1 - e^{-fV} \sum_{i=0}^{k-1} \frac{(fV)^i}{i!} \right], \quad k = 1, 2, \dots;$$

$$P_{loss} = \begin{cases} \frac{1-\rho}{e^{(1-\rho)fV} - \rho}, & \text{if } \rho \neq 1, \\ (1+fV)^{-1}, & \text{if } \rho = 1. \end{cases} \quad (5)$$

B. System $M/G/1/\infty$ with unlimited buffer space and service time dependent on customer's volume

Consider a system $M/G/1/\infty$ with unlimited buffer space and arbitrary joint DF $F(x, t)$ of RVs ζ and ξ . Let $\alpha(s, q) = \int_0^{\infty} e^{-sx-qt} dF(x, t)$ be a double Laplace-Stieltjes transform (LST) of DF $F(x, t)$. Then, $\varphi(s) = \alpha(s, 0) = \int_0^{\infty} e^{-sx} dL(x)$ and $\beta(q) = \alpha(0, q) = \int_0^{\infty} e^{-qt} dB(t)$ are LSTs of DFs $L(x)$ and $B(t)$, respectively. Let $\beta_1 = E\xi$ be the mean value of service time. Assume that $\rho = a\beta_1 < 1$, where a is parameter of arrival process. Then, for the system under consideration, finite steady-state characteristics exist. E.g. we have $\sigma(t) \Rightarrow \sigma$ and $\eta(t) \Rightarrow \eta$ in the sense of a weak convergence, where σ and η are the steady-state total customers' volume and number of customers in the system, respectively. Let $D(x)$ be DF of RV σ . Denote by $\delta(s)$ its LST. Then, we obtain [7–9]:

$$\delta(s) = \int_0^{\infty} e^{-sx} dD(x) = (1-\rho) \left[1 + \frac{\varphi(s) - \alpha(s, a - a\varphi(s))}{\beta(a - a\varphi(s)) - \varphi(s)} \right].$$

Using the last relation, we can calculate moments of RV σ (if they exist). For the first and second ones, we have [8, 9]:

$$\delta_1 = E\sigma = -\delta'(0) = a\alpha_{11} + \frac{a^2\beta_2\varphi_1}{2(1-\rho)};$$

$$\delta_2 = E\sigma^2 = \delta''(0) = a(\alpha_{21} + a\varphi_1\alpha_{12}) + \frac{a^3\beta_2\varphi_1\alpha_{11}}{1-\rho} + \frac{a^2\beta_2\varphi_2}{2(1-\rho)} + \frac{a^3\beta_3\varphi_1^2}{3(1-\rho)} + \frac{a^4\beta_2^2\varphi_1^2}{2(1-\rho)^2},$$

where φ_i, β_i are the i th moments of RVs ζ and ξ , respectively, $i = 1, 2, \dots$, α_{ij} is the mixed $(i + j)$ th moment of the random vector (ζ, ξ) , $i, j = 1, 2, \dots$.

Assume e.g. that customer's volume ζ has an exponential distribution with parameter f , and service time of the customer is proportional to his volume ($\xi = c\zeta$, $c > 0$). In this case, we can determine an explicit form of DF $D(x)$ [9]:

$$D(x) = 1 + \frac{\rho^2 e^{-(1-\rho)fx}}{1-2\rho} - \frac{\rho(1-\rho)}{\sqrt{\rho(4+\rho)}} \left(\frac{1-b_1}{1-b_1-\rho} e^{-b_1fx} - \frac{1-b_2}{1-b_2-\rho} e^{-b_2fx} \right), \quad \text{if } \rho \neq 1/2,$$

$$D(x) = 1 + \frac{1}{9} e^{-2fx} - \frac{1}{3} \left(\frac{11}{6} + \frac{fx}{4} \right) e^{-fx/2}, \quad \text{if } \rho = 1/2,$$

where $\rho = ac/f$,

$$b_1 = \frac{2+\rho-\sqrt{\rho(4+\rho)}}{2}, \quad b_2 = \frac{2+\rho+\sqrt{\rho(4+\rho)}}{2}. \quad (6)$$

Note that, in the case of exponential distribution of RV ζ and the same dependence between ζ and ξ , we can also determine an explicit form of $D(x)$ for the system $M/M/n/m$ with

unlimited buffer space, $n = 1, 2, \dots, m = 0, 1, \dots$ [10]. E.g. for $n = 2, m = \infty, \rho = ac/(2f) < 1$, we obtain:

$$D(x) = 1 + \frac{2\rho e^{-(1-\rho)fx}}{(1-\rho)(1-2\rho)} - \frac{2\rho^2 p_0}{\sqrt{\rho(4+\rho)}} \times$$

$$\times \left[\frac{1-b_1}{(1-b_1)(1-b_1-\rho)} e^{-b_1fx} - \frac{1-b_2}{(1-b_2)(1-b_2-\rho)} e^{-b_2fx} \right],$$

if $\rho \neq 1/2$,

$$D(x) = 1 - \frac{2}{27} [(3fx+10)e^{-fx/2} - e^{-2fx}], \text{ if } \rho = 1/2,$$

where $p_0 = (1-\rho)/(1+\rho)$ and b_1, b_2 are calculated by relations (6).

IV. LOSS CHARACTERISTICS FOR QUEUEING SYSTEMS WITH RANDOM VOLUME CUSTOMERS

Let $D(x)$ be steady-state DF of the total customers' volume in a queueing system with limited (by V) buffer space ($D(x) = 1$, if $x > V$).

The most familiar loss characteristic is loss probability P_{loss} , i.e. the relative part of number of demands that were lost in steady state during infinite time interval. For the system with Poisson arrival process, in which only buffer space is limited and there are no other limitations, we can determine P_{loss} by the following relation [9, 11]:

$$P_{loss} = 1 - \int_0^V D(V-x) dL(x), \tag{7}$$

where $L(x)$ is DF of RV ζ .

More generally, P_{loss} can be obtained from the stability conditions, i.e. from the fact that (in steady state) the mean number of customers accept to the system during some time interval must be equal to the mean number of customers complete their service during the same interval, as it was done when the system $M/M/1(\infty, V)$ was analyzed.

It is clear that P_{loss} is the most popular loss characteristic, but it is not very objective for the systems under consideration, because of difference between probabilities to be lost for customers of different volumes.

Other (may be more objective) loss characteristic is the probability Q_{loss} of volume unit losing [9, 11], i.e. the relative part of total volume of customers that were lost during infinite time interval. For the system with Poisson arrival process and without other limitations except of buffer space one, this probability can be calculated by the following relation:

$$Q_{loss} = 1 - \frac{1}{\varphi_1} \int_0^V x D(V-x) dL(x),$$

where $\varphi_1 = E\zeta = \int_0^\infty x dL(x)$ is the mean customer's volume.

Generally, Q_{loss} can be also obtained from the stability conditions, i.e. from the fact that (in steady state) the mean total volume of customers accept to the system during some time interval must be equal to the mean total volume of customers complete their service during the same interval. However, in this case, appropriate calculations are significantly more complicated than in the case of P_{loss} ones. We can easily

prove that $Q_{loss} \geq P_{loss}$, if these characteristics were calculated by relations (7) and (8). More generally, we can obtain this inequality using the theory of regenerative processes [12, 13].

Assume that the processes $\sigma(t)$ and $\eta(t)$ are regenerative with finite mean value of regeneration cycle in the system under consideration. Let n_j be the number of arriving customers during j th regeneration cycle, n_j^* be the number of losing ones during this cycle, $j = 1, 2, \dots, N$, v_{ij}^* be the volume of i th losing customer within this cycle, $i = 1, 2, \dots, n_j^*$, V_j^* be the total losing volume within this cycle: $V_j^* = \sum_{i=1}^{n_j^*} v_{ij}^*$.

As it follows from the theory of regenerative processes [13], we have $N^{-1} \sum_{j=1}^N V_j^* \xrightarrow{a.s.} E\chi^*$ when $N \rightarrow \infty$, where RV χ^* is the total volume of losing customers within arbitrary regeneration cycle, $N^{-1} \sum_{j=1}^N n_j^* \xrightarrow{a.s.} Ev^*$, where RV v^* means the number of losing customers during a regeneration cycle. It is clear that $E\chi^* = Ev^* E(\zeta|A)$, where $E(\zeta|A)$ is a conditional mean value of customer's volume under condition that the customer is lost (event A). By similar way, we obtain that $N^{-1} \sum_{j=1}^N n_j \xrightarrow{a.s.} Ev$, where RV v means the number of arriving customers on arbitrary regeneration cycle. Then, $E\zeta \cdot Ev$ is the mean total volume of arriving customers on arbitrary cycle and we obtain:

$$\frac{N^{-1} \sum_{j=1}^N V_j^*}{E\zeta Ev} \xrightarrow{a.s.} Q_{loss} = \frac{E(\zeta|A)}{E\zeta} \cdot \frac{Ev^*}{Ev} = \frac{E(\zeta|A)}{E\zeta} P_{loss}.$$

Hence, $Q_{loss} \geq P_{loss}$, if $E(\zeta|A) \geq E\zeta = \varphi_1$, i.e. the mean volume of losing customers is equal or greater than the mean volume of arbitrary ones. It is clear that $Q_{loss} = P_{loss}$, if the loss of a customer does not depend on his volume.

It was proved [14] that the relation $P_{loss} = 1 - (1 - p_0)/\rho$ holds for one-server systems when service time does not depend on customer's volume, and the relation $Q_{loss} = 1 - (1 - p_0)/\rho$ holds when service time is proportional to customer's volume.

We can obtain loss characteristics in explicit form for some queueing systems with random volume customers. E.g. in Part III of the paper, we obtained probability P_{loss} for system $M/M/1(\infty, V)$ with independent RVs ζ and ξ , when RV ξ is distributed exponentially (see relation (5)). The same formula takes place for processor sharing system with the same customer's volume distribution and customer's length [15] independent of his volume that has exponential distribution with parameter f . For these systems, we can obtain the following relation for Q_{loss} [11]:

$$Q_{loss} = \frac{p_0 e^{-fV}}{\rho} [(1+\rho)e^{\rho fV} - 1],$$

where p_0 is determined by relation (4).

Assume that, in processor sharing system, customer's volume has exponential distribution with parameter f and customer's length is proportional to his volume ($\xi = c\zeta$, $c > 0$). Then, we obtain [11]:

$$P_{loss} = p_0 e^{-fV} \cosh(\sqrt{\rho} fV),$$

$$Q_{loss} = \frac{p_0 e^{-fV}}{\sqrt{\rho}} [\sinh(\sqrt{\rho} fV) + \sqrt{\rho} \cosh(\sqrt{\rho} fV)],$$

where $\rho = ac/f$,

$$p_0 = \begin{cases} \frac{1-\rho}{1-\sqrt{\rho} e^{-fV} [\sinh(\sqrt{\rho} fV) + \sqrt{\rho} \cosh(\sqrt{\rho} fV)]}, & \text{if } \rho \neq 1, \\ \frac{4}{3+2fV+e^{-2fV}}, & \text{if } \rho = 1. \end{cases}$$

V. ESTIMATION OF LOSS CHARACTERISTICS IN QUEUEING SYSTEMS WITH DEPENDENT CUSTOMER'S VOLUME AND SERVICE TIME

For models with $V < \infty$ and dependent RVs ζ and ξ (see Part III), we, as a rule, cannot exactly determine loss characteristics. An exception are some models without waiting places, e.g. $M/G/n/(0, V)$ or processor sharing systems [16, 17]. But, in this case, we often can estimate this characteristics with the help of models with unlimited buffer space.

Let QS_∞ be the system with Poisson arrival process and without any limitations (there are no losses in it). Let QS_V be the system, which differs from QS_∞ in limitation (by V , $V < \infty$) of buffer space capacity, only. Assume that steady state exists for both systems. Denote by $D_\infty(x)$ and $D_V(x)$ DFs of the steady-state total volume in systems QS_∞ and QS_V , respectively.

It was proved [11] that inequality $D_\infty(x) \leq D_V(x)$ holds for all real x . Then, we obtain the following inequalities:

$$P_{loss} = 1 - \int_0^V D_V(V-x) dL(x) \leq 1 - \int_0^V D_\infty(V-x) dL(x) = P_{loss}^*,$$

$$Q_{loss} = 1 - \frac{1}{\phi_1} \int_0^V x D_V(V-x) dL(x) \leq$$

$$\leq 1 - \frac{1}{\phi_1} \int_0^V x D_\infty(V-x) dL(x) = Q_{loss}^*.$$

As it follows from these inequalities, the values P_{loss}^* and Q_{loss}^* can be interpreted as an upper boundaries for P_{loss} and Q_{loss} , respectively, when DF $D_\infty(x)$ is known.

Unfortunately, calculation of these characteristics is often very complicated, because we have function $D_\infty(x)$ very rarely in exact form. For this function approximation, we often can use incomplete Gamma function $\gamma(p, bx)$ and Gamma function $\Gamma(p) = \gamma(p, \infty)$. E.g. in both inequalities, we sometimes can approximate $D_\infty(x)$ by the function [9]:

$$D_\infty^*(x) = p_0 + (1-p_0) \frac{\gamma(p, bx)}{\Gamma(p)}.$$

The values of parameters p and b we choose so that the first and second moments of the approximate distribution are equal to the first (δ_1) and second (δ_2) moments of DF $D_\infty(x)$. Then, we have:

$$p = \frac{\delta_1^2}{(1-p_0)\delta_2 - \delta_1^2}, \quad b = \frac{(1-p_0)\delta_1}{(1-p_0)\delta_2 - \delta_1^2}.$$

E.g., if RV ζ is absolute continuous, we obtain for Q_{loss}^* the following approximate relation [18]:

$$Q_{loss}^* \approx 1 - \frac{1}{\phi_1} \int_0^V x \left[p_0 + (1-p_0) \frac{\gamma(p, b(V-x))}{\Gamma(p)} \right] l(x) dx,$$

where $l(x)$ is density of RV ζ . To calculate P_{loss}^* and Q_{loss}^* , we can use methods of numerical integration (see e.g. [19]).

VI. AN ANALOG OF LITTLE'S FORMULA

Little's formula $E\eta = \lambda E\theta$, where η is number of customers present in the system in steady state, λ is intensity of (stationary) arrival process, θ is sojourn time of a customer in the system, is one of the most significant result of classical queueing theory. It holds for arbitrary steady-state system without losing of customers.

Consider an arbitrary queueing system with random volume customers. Assume that RV ζ means customer's volume and ξ is its service time. Denote by $E(\zeta\xi)$ the mixed moment of (1+1)th order of RVs ζ and ξ . Assume that arrival process is stationary with intensity λ . Let ω be queueing time of a customer, hence, $\theta = \omega + \xi$, where RVs ω and ξ are independent. Let σ be the total volume of customers present in the system in steady state. It was proved (see [20, 21]) that the following relation named an analog of Little's formula holds:

$$E\sigma = \lambda(E\omega \cdot E\xi + E(\zeta\xi)). \quad (9)$$

Consider, as an example, two steady-state systems of $M/M/1/\infty$ type: QS_1 and QS_2 . Let a be parameter (intensity) of arrival process in both systems. Assume that customer's volume is distributed exponentially with parameter f in each of them, i.e. $L(x) = P\{\zeta < x\} = 1 - e^{-fx}$. Assume that customer's volume and service time are independent in QS_1 . Denote by μ parameter of service time in it, i.e. $B_1(t) = 1 - e^{-\mu t}$. Assume that, in QS_2 , service time is proportional to customer's volume: $\xi = c\zeta$, $c > 0$. Then, we have for this system:

$$B_2(t) = P\{\xi < t\} = P\{c\zeta < t\} = P\{\zeta < t/c\} = 1 - e^{-ft/c}.$$

Assume that the equality $\mu = f/c$ holds. Then, we obtain $B_1(t) = B_2(t)$. It means that the systems QS_1 and QS_2 are identical from classical queueing theory point of view, i.e. all

classical characteristics (e.g. queueing time, sojourn time, number of customers present in the system) are the same in both systems. But what about $E\sigma$ ($E\sigma^{(1)}$ and $E\sigma^{(2)}$ for systems QS_1 and QS_2 , respectively)?

RVs ζ and ξ are independent in QS_1 . Then, for their mixed moment $\alpha_{11}^{(1)}$, we obtain:

$$\alpha_{11}^{(1)} = E(\zeta\xi) = E\zeta \cdot E\xi = (f\mu)^{-1}.$$

Classical characteristic $E\omega$ is the same for both systems. It is known [5, 12] that $E\omega = \rho/(\mu(1-\rho))$, where $\rho = a/\mu = ac/f$. Then, for QS_1 , we have from relation (9):

$$E\sigma^{(1)} = a \left[\frac{\rho}{\mu(1-\rho)} \cdot \frac{1}{f} + \frac{1}{f} \cdot \frac{1}{\mu} \right] = \frac{1}{f} \cdot \frac{\rho}{1-\rho}.$$

For mixed moment $\alpha_{11}^{(2)}$ of the system QS_2 , we have:

$$\alpha_{11}^{(2)} = E(\zeta\xi) = E(\zeta \cdot c\xi) = cE\xi^2.$$

For the second moment $E\xi^2$ of customers' volume, we easily obtain $E\xi^2 = 2/f^2$ and finally have:

$$E\sigma^{(2)} = a \left[\frac{\rho}{\mu(1-\rho)} \cdot \frac{1}{f} + \frac{2c}{f^2} \right] = \frac{1}{f} \cdot \frac{\rho(2-\rho)}{1-\rho}.$$

Therefore, mean total volume is $2-\rho$ times greater in the system QS_2 . It is intuitively clear, because customers of a greater volume are served for a longer time in this system.

This simple example shows us that we need to take into account the type of dependence between RVs ζ and ξ , if we want correctly determine buffer space capacity when we design a communicating center (router) of a computer or telecommunication net.

VII. CONCLUSION

In the paper, we present a class of queueing systems with non-homogeneous (or random volume) customers that are the models, which can be used for designing of nodes in computer and communicating networks when we need to determine buffer space capacity of the node.

We classify models within the theory, analyze its basic principles and performance characteristics and present some simple examples of queueing models with random volume customers.

We also discuss the problem of calculation and estimation of loss characteristics in systems with random volume customers. In addition to ordinary loss probability that characterizes the part of losing customers, we present probability of volume unit losing and consider ratio of these characteristics.

We present also an analog of Little's formula. It is the general relation that shows a correlation between mean total volume of customers and (1+1)th mixed moment of customer's volume and his service time. As a result, we come to conclusion that *we have to take into account the dependence between customer's volume and service time when determine*

buffer space capacity of the node of computer or communication network. Substantiation of this statement is the basic aim of our paper.

REFERENCES

- [1] V. N. Roginsky, A. D. Kharkevich, and M. A. Shneps, Communication Networks Theory, Moscow: Radio i Svyaz, 1981 (in Russian).
- [2] M. Schwartz, Computer-Communication Network Design and Analysis, New Jersey: Prentice-Hall, 1977.
- [3] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison-Wesley Publishing Company, 1987.
- [4] N. B. Zeliger, O. S. Chougreev, and G. G. Yanovsky, Design of Networks and Systems of Discrete Messages Transmission, Moscow: Radio i Svyaz, 1984 (in Russian).
- [5] P. P. Bocharov, C. D'Apice, A. V. Pechinkin, and A. V. Salerno, Queueing Theory, Utrecht-Boston: VSP, 2004.
- [6] A. M. Alexandrov and B. A. Kaz, "Non-homogeneous demands flow service", Izvestiya AN SSSR. Technicheskaya Kibernetika, no. 2, 1973, pp. 47–53 (in Russian).
- [7] B. Sengupta, "The spatial requirements of an $M/G/1$ queue, or: how to design for buffer space", In: F. Baccelli and G. Fayolle (eds.) Modeling and Performance Evaluation Methodology, LNCS, vol. 60, Springer, Heidelberg, 1984, pp. 547–562.
- [8] O. M. Tikhonenko, Queueing Models in Computer Systems, Minsk: Universitetskoe, 1990 (in Russian).
- [9] O. Tikhonenko, Computer Systems Probability Analysis, Warsaw: Akademicka Oficyna Wydawnicza EXIT, 2006 (in Polish).
- [10] O. Tikhonenko and M. Kawecka, "Total volume distribution for multiserver queueing systems with random capacity demands", In: A. Kwiecień, P. Gaj, and P. Siera (eds.) Computer Networks 2013, CCIS, vol. 370, Berlin-Heidelberg: Springer-Verlag, 2013, pp. 394–405.
- [11] O. Tikhonenko, "Determination of loss characteristics in queueing systems with demands of random space requirement", In: A. Dudin et al. (eds.) Information Technologies and Mathematical Modelling, CCIS, vol. 564, Springer International Publishing Switzerland, 2015, pp. 209–215.
- [12] S. Asmussen, Applied Probability and Queues, 2d ed., New York: Springer-Verlag, 2003.
- [13] L. Lakatos, L. Szeidl, and M. Telek, Introduction to Queueing Systems with Telecommunication Applications, New York: Springer, 2010.
- [14] E. Morozov, R. Nekrasova, L. Potakhina, and O. Tikhonenko, "Asymptotic analysis of queueing systems with finite buffer space", In: A. Kwiecień, P. Gaj, and P. Siera (eds.) Computer Networks 2014, CCIS, vol. 431, Springer International Publishing Switzerland, 2014, pp. 223–232.
- [15] S. F. Yashkov and A. S. Yashkova, "Processor sharing: a survey of the mathematical theory", Automation and Remote Control, vol. 68, no. 9, 2007, pp. 1662–1731.
- [16] O. M. Tikhonenko, "Generalized Erlang problem for service systems with finite total capacity", Problems of Information Transmission, vol. 41, no. 3, 2005, pp. 243–253.
- [17] O. M. Tikhonenko, "Queueing systems with processor sharing and limited resources", Automation and Remote Control, vol. 71, no. 5, 2010, pp. 803–815.
- [18] O. Tikhonenko and M. Ziółkowski, "Single-server queueing system with external and internal customers", Bulletin of the Polish Academy of Sciences, Technical Sciences, vol. 66, no. 4, 2018, pp. 539–551.
- [19] Y. L. Luke, Mathematical Functions and their Approximations, New York: Academic Press Inc., 1975.
- [20] O. M. Tikhonenko, "An analog of Little's formula for queueing systems with nonuniform customers", Automation and Remote Control, vol. 57, no. 1, 1996, pp. 85–88.
- [21] E. Morozov, L. Potakhina, and O. Tikhonenko, "Regenerative analysis of a system with a random volume of customers", In: A. Dudin et al. (eds.) Information Technologies and Mathematical Modelling, CCIS, vol. 638, Springer International Publishing Switzerland, 2016, pp. 261–272.

Automatic restoration of diacritics based on word n-grams for Slovak texts

Štefan Toth, Emanuel Zaymus, Michal Ďuračik, Matej Meško, Patrik Hrkút

Department of Software Technologies
Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia

{stefan.toth, michal.duracik, matej.mesko, patrik.hrkut}@fri.uniza.sk, emanuel.zaymus@gmail.com

Abstract—In the past and even now, many people still write texts without diacritics, especially in chat messages, e-mails or discussion posts. This issue evolved from historical reasons when people had a problem with text encoding in messages or wanted to write them faster. In this paper, we propose an algorithm based on word n-grams (contiguous sequence of n words) that restore diacritics of text written in the Slovak language. We also compare and evaluate our results with existing algorithms developed for Slovak texts.

Keywords—*diacritic, diacritics restoration, n-gram, Slovak language*

I. INTRODUCTION

A diacritic, defined by Merriam-Webster dictionary [1], is a mark near or through an orthographic or phonetic character or combination of characters indicating a phonetic value different from that given the unmarked or otherwise marked element. Another simple definition is that a diacritic is a mark that is placed over, under or through a letter to show that the letter should be pronounced in a particular way.

Some people miss diacritical marks (diacritics) when typing, especially in instant chat communication, where users send to each other a lot of short messages by messengers, SMS or e-mails. This skipping is developed due to a need for a quick response or minimalization of time to send such a message. Practically this means to write a message as fast as possible, where diacritics omitting is an acceptable error.

There can also be a historical cause of this. In the past, character encoding was a problem. Various mail client applications have used different encodings for texts written in the specific language on different operating systems. It happened that a received mail from one mail client was not displayed correctly on another client. Therefore, people used to write texts without diacritics. When the era of mobile phones came, one of the main communication channels has become short text messages (called SMS by Short Message Service). The maximum length of one SMS is 140 bytes [2]. If the sender uses basic Latin characters without diacritics in a message, the default encoding is 7-bit (GSM-7). This means that the message can hold 160 characters. However, if a user uses non-standard or accented characters, the number of characters in a message will be truncated due to a change in the encoding (from 7-bit to 8bit or currently even 16bit for UCS2). For this reason, people preferred to write SMS messages without diacritics.

In sum up, omitting diacritics was a purely economic solution. But people soon realize, that this makes typing messages faster, too. Nowadays, the size of a message is not a problem (because of good access to the internet) and intelligent auto-correction can help to correct diacritic. But still, some people do not use diacritical marks in typing.

Reason for this may be quick writing, habit, convenience, laziness or ignorance typing.

Occasionally, the contents of a mail or an SMS need to be used in another text. If its text is written without diacritics, then it is necessary to either manually overwrite or semi-automatically correct the text by using the spell checker. This is quite a laborious solution. To simplify and speed up the transformation, an automatic restoration system of diacritics would be a suitable solution.

In this paper, we are focusing on the Slovak language and its specific diacritical marks. Table I shows all lowercase and uppercase letters in the Slovak alphabet.

TABLE I. DIACRITICS IN THE SLOVAK LANGUAGE

Diacritical mark	Slovak letters with diacritical marks
´ (acute accent)	á, é, í, ľ, ó, ŕ, ú, ý Á, Ě, Ľ, Ó, Ŕ, Ú, Ý
˘ (caron)	č, ď, dž, ň, ř, š, ť, ž Č, Ď, DŽ, Ň, Ř, Š, Ť, Ž
¨ (diaeresis)	ä Ä
ˆ (circumflex accent)	ô Ô

Text can also contain diacritical marks that are no native to the current language, because there can be used words from foreign languages, containing letters such as: ö, ü, ù, Ö, Ü, Ů, ā, Ā, ē, Ě, ê, Ê, é, Ě, ě, Ć, ś, Ś, ç, Ç, ș, Ș, ț, Ț, ... [3].

If a word without diacritics has only one meaning (there are no other meaning acquired by adding diacritics), the reconstruction can be done using a dictionary. Example of such word in the Slovak language is “bábätko” (translation: “a baby”). Removing diacritics, we obtain the word “babatko” as a result. There is no other word with other meaning in the Slovak language that can be created by adding a combination of diacritical marks. So, in this case, it is easy to restore correct diacritics.

The more difficult problem is a word which can have different meaning adding a various combination of diacritics. Example of such word is “boli” (meaning “we were” or “they were”) and itself is a valid word. One of meaning we can reconstruct from this word is “bolí” (meaning “it hurts”) and another one is “bôli” (meaning “in a grief”). Here must be the context understand to reconstruct diacritics correctly.

II. CURRENT STATE

Several works have been published for the restoration of diacritics for the Slovak language. For example, in [4] authors used *Hidden Markov Model* and *Viterbi algorithm*. In [5] authors used *statistical language models* of Slovak language. Nowadays, *deep neural networks* have shown remarkable results as authors in [6] and [7] show state of the art solutions.

Reconstruction of diacritics is not only important for the Slovak language. There are many languages where diacritical marks are used. Publications dealing with this problem can be found for languages such as Czech [7], Hungarian [8], Romanian [9], Arabic [10], Vietnamese [11], and many others.

Now, we are mainly focusing on the Slovak language. There are three freely accessible services for the reconstruction of diacritics on the internet. These services will be described in the following sections.

A. Diakritik by SAS

The tool *Diakritik* was created by Ľ. Štúr Institute of Linguistics of the Slovak Academy of Sciences (SAS). It was open for public by 18.8.2014 [12]. It is based on a big corpus of Slovak text and has several options in how diacritics is reconstructed: first match, random match, most used and words n-gram (2, 3, 4, 5 or 6 words) and more. Authors of the Diakritik announce that it has a 0.21% error rate.

B. Statistical diacritic restoration by STUBA

The *statistical diacritic restoration* system was created by J. Geder as bachelor thesis [13] at the Faculty of Informatics and Information Technologies at the Slovak University of Technology in Bratislava (STUBA). The application creates all possible combination of n-words with correct diacritics. Next step calculates the probability of occurrence for all n-words combinations and one with the highest probability is chosen. The algorithm can reconstruct only words without a typo or missing letters. The announced success rate is 98%.

C. Diacritics restoration by BRM

The *diacritics restoration* (in Slovak “Dopĺňač diakritiky”) (BRM) by R. Hraška is a web application open to the public [14]. The restoration is determined by most often word from diacritics form. The application does not use any sophisticated methods or algorithms or complicated linguistics methods.

III. PROPOSED ALGORITHM

Our algorithm is based on comparing the surrounding of the word to which we want to add diacritics with the n-grams associated with that word. We consider the words around 3 words before and 3 words after this word in context, if these words exist. The base algorithm scheme is shown in Fig. 1.

After reading the input text without diacritics, which is the text we want to reconstruct diacritics, we divide the text into separate words. In this step, we need to keep in mind that in our final text we want to preserve all the punctuation, white, and other characters that the user enters. Now we will process all the words one by one. If we still have a word, we take it and normalize it (convert it into a standardized form that we can process).

Let's convert all capital letters to lowercase. We do this because all words in n-grams are only lowercase, which simplifies the work and saves a lot of memory (if we were to work with data that also distinguishes between uppercase and lowercase letters, processing and searching such data would be much more complicated). We get the neighborhood word. When comparing, we will use no more than 4-grams. If the first word in a 4-gram is equal to our searching word, we need 3 following words after our word for comparison. If the last word in the 4-gram is equal to our word, we need 3 words before our word for comparison. Therefore, we consider the

word context around 3 words before and 3 words after this word, of course, if these words exist.

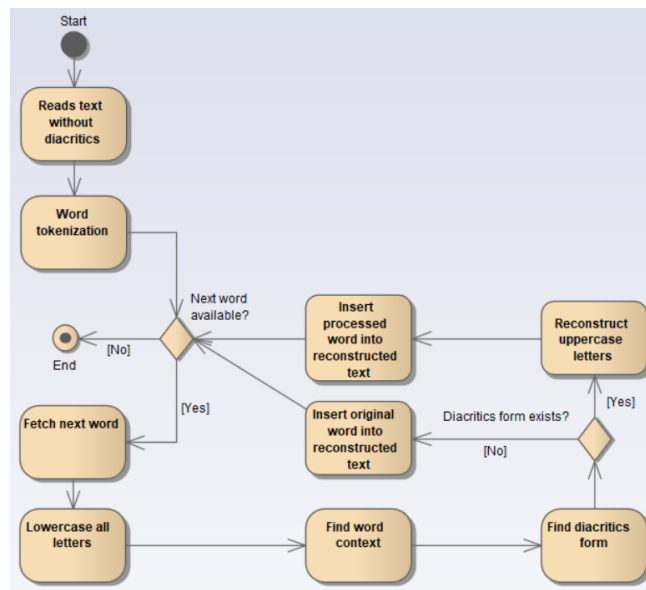


Fig. 1. Activity diagram – base algorithm scheme

Subsequently, we find diacritics form of the word (more information will be described in the next section, see Fig. 2). If we have successfully added diacritical marks, we change the lowercase letters to uppercase letters in the new word as they were in the original word without diacritics and write it into the resulting text. However, if we were not successful, we will add the original word to the resulting text, which we do not need to change to lowercase. When inserting words into the resulting text, we need to be careful to insert punctuation, white, and other characters correctly to preserve the original format of the text. Consequently, if we have another word, we begin by processing it, if not, our algorithm ends.

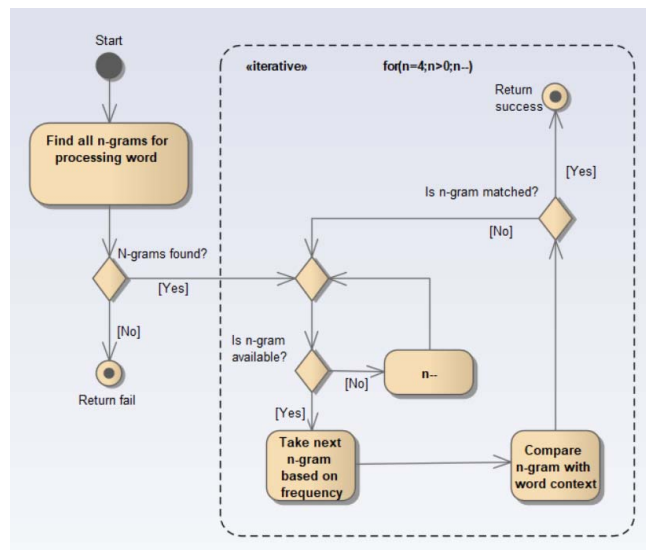


Fig. 2. Activity diagram – matching n-grams by the surrounding context

In the algorithm to find diacritics form of a word using its surroundings (see Fig. 2) we assume that we have a word for the restoration of diacritics and its surroundings. First, we find all the n-grams for processing word. By the corresponding n-grams, we mean every n-gram in which at least one of the n words corresponds to our reconstructed

word (of course, if we remove the diacritics from the n -gram before comparing them). If we do not find any such n -gram, we end with returning fail, that is, we have not added any diacritics to that word. If we find at least one such n -gram, we can proceed to their gradual processing. In the above diagram, we see the for-cycle for ($n = 4; n > 0; n--$), which says that we will gradually process groups of the corresponding n -grams, ranging from 4-grams to 1-grams. If there is no n -gram in the current group, we decrement n . In this way, we iterate from the group of n -grams to the group of $(n-1)$ -grams while $n > 0$.

If we have some n -gram in the current group, we take the one that has the highest absolute frequency. In the n -gram, we find where our word occurs (we have to keep in mind that the searched word may appear multiple times in the n -gram), and we compare the surrounding words of the n -gram with the surroundings of the reconstructed word (of course, we remove the diacritics from the n -gram before processing).

If we find a match, that is, the surrounding of the word agrees with the n -gram, we can return success and reconstruct diacritics by the found word from the n -gram. If not, let's see if we still have any n -gram in the current n -gram group. We can notice the fact that if we go into the for-cycle, our only possible exit is a return to success. This is because we assume that in the group of n -grams belonging to a given word there is necessarily at least one 1-gram. This condition ensures that if the cycle goes to 1-gram comparison, immediately at the first 1-gram, the cycle returns success because 1-gram has no neighborhood, and hence the match is certain.

If we find a match around a word with an n -gram with a higher n , we consider it more relevant than a match in a lower n , even if that n -gram had a higher absolute frequency. That's why we first search all 4-grams by frequency and then all 3-grams by frequency.

To help us quickly search through all the data associated with each word, we used the Trie data structure, which has an $O(n)$ search complexity where n is number of items.

Because it is necessary to compromise between speed and accuracy (success) of the algorithm, we decided to implement two different ways, one focusing on speed (data structure in memory) and the other focusing on success (data structure on disk). It will always be crucial how and where we store the data (n -grams) that we want to search.

A. The data structure in memory

Implementation of the data structure in main memory is very simple. We follow the algorithm design as shown in Fig. 2, and when we find the *Find all n -grams for processing word*, the above-mentioned Trie structure comes into play. We chose the easiest approach – each word will have its corresponding n -grams at the top of the Trie tree. This ensures almost instant access to all necessary n -grams. In addition, the format of storing data in vertices can be adjusted so that we do not have to perform a for-cycle at all, nor do we search for the next n -gram from the *Take next n -gram based on frequency*. This means that all relevant 4-grams, then 3-grams, 2-grams and finally 1-grams will be on the leaf first. If we have n -grams stored in this memory, we don't have to hold information about their frequencies at all (which saves us memory). In result, the algorithm for finding n -gram is shown in Fig. 3.

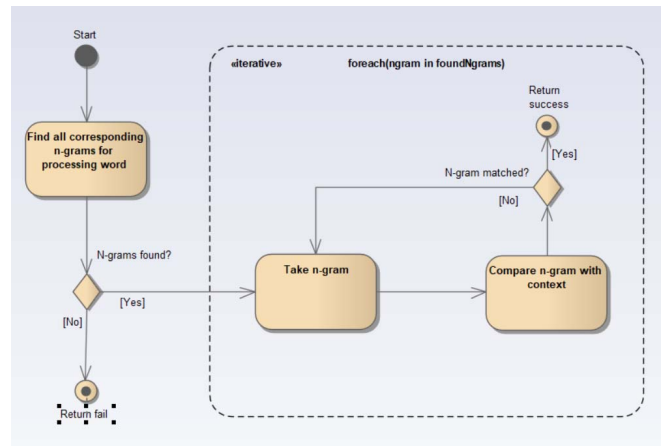


Fig. 3. Activity diagram – n -gram diacritic search using Trie

Based on our experiments, we can optimize the algorithm as follows:

1. There is no need to keep words without diacritics. They will be not found in the structure and skipped.
2. If there is exactly one word with diacritics for one word without diacritics, we can only store it in 1-gram database and remove all n -grams containing the word.
3. 1-gram cannot have multiple diacritics variations, because there is no context to use for the match. So, 1-gram contains only one the most often form.

We found, that the number of used n -grams are always approached in ratios showed in Table II.

TABLE II. PERCENTUAL AMOUNT OF N -GRAMS

	4-grams	3-grams	2-grams	1-grams
Percentual use of diacritics reconstruction	10%	24%	37%	29%

Table II shows the percentage of n -grams in a test database that was used in diacritics reconstruction. Using the memory only makes searching and whole algorithm very fast, but we want out to be runnable on common PCs (average 8 GB). We estimated, that we must not exceed 4-5 GB of RAM to be runnable on such PC. To approximately reach this limit maximum number of n -grams must be limited to 700 per one word. Limit of 700 n -grams involves all n -grams levels with percentual distribution described in Table II.

B. The data structure in disk

Our memory restriction makes need to use a disk to store data of n -grams. Adding this possibility greatly increases the size of the potential database. Using disk as storage for n -grams creates minor changes in our structure. Trie nodes contain the only reference to a binary file stored on disk as shown in Fig. 4. More in detail, nodes in Trie will have one reference to index in the binary file, the p position. This position in the file contains the count of n -gram k , which is a 4-byte number (integer). Next, all n -grams of p position are processed. N -grams in the binary file are also ordered by its occurrence frequency and by the count of n -grams.

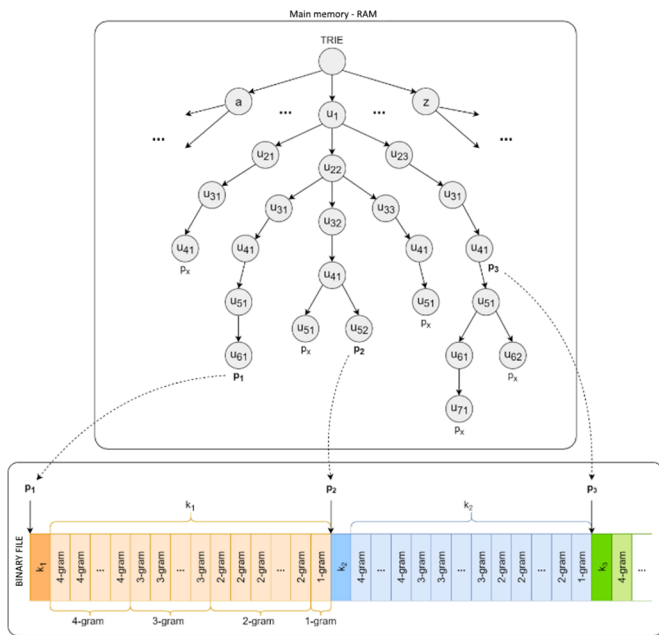


Fig. 4. The structure of n-gram database using the binary file

During our early test, we notice showiness of our implementation algorithm implementation as very slow. The issue was created by an enormous count of n-grams of one word reaching millions in some cases of word with high occurrence frequency. Our analysis provides, that most common n-gram count for one word is between 30,000-40,000. Based on our observation we set a limit of maximum 40,000 n-grams per word. This limitation speeds reconstruction of one word from 0.27s to 0.009s and success rate was decreased negligibly.

Another optimization step was in using cache memory for frequently repeating words or in a similar context. In our implementation, the cache stores last 1000 searched words with picked n-grams.

IV. DATA

For building a good n-gram database is essential to use quality source. Our data was created from public-open texts from Slovak national corpus from the Slovak Academy of Sciences [15]. The corpus contains a primary large number of words from a huge range of categories (many literary genres, regions, fields of science, etc.) from the year 1995. All words in the corpus contain additional information about them generated by *IRSTLM Toolkit* [16]. Although we are aware that this data also contains incorrect n-grams, there is a very large number of them. We have used a version of the corpus labelled as *prim-8.0* released at 31.01.2018 with the token count at 1.5bln. For our algorithm, we are using sub-corpus *prim-8.0-public-all*.

V. EXPERIMENTS

To verify the success of the proposed algorithms, we chose a set of our test texts. We divided them into three categories according to literary styles: professional, journalistic and artistic texts. After randomly selecting 25 texts by category, we obtained a set of more than 53,000 words, each category having approximately 17,000-18,000 words.

Subsequently, we removed diacritics from these texts and performed tests with our and mentioned algorithms. After

that, we compared the reconstructed words with the original words. The overall results of the tests show Table III and Fig. 5.

TABLE III. RESULTS OF ALGORITHMS ON OUR DATASET SORTED BY SUCCESS RATE

Algorithm	Error rate	Success rate
Our algorithm (disk)	1.83%	98.17%
Our algorithm (memory)	1.93%	98.07%
BRM	3.00%	97.00%
SAS	3.05%	96.95%
STUBA	4.53%	95.47%

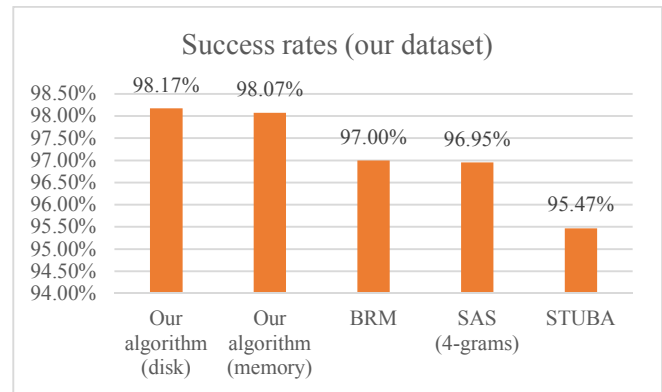


Fig. 5. The chart of overall success rates of the algorithms on our dataset

Also, we chose another experiment in which we used part of the dataset from the paper by Náplava et al. [7] published on [17]. Specifically, we chose a test set for the Slovak language obtained from Wikipedia articles that contains 524,694 words.

Since individual web diacritics restoration algorithm services have limitations on the length of input text, we have divided the test part of the dataset into subsets with words of max. 9000 characters. We reached 369 subsets, but only 350 of them were used because we had trouble getting the results from the STUBA algorithm. The reasons were unhandled exceptions and errors with empty results on 19 subsets of texts. Nevertheless, we further found that this reconstructor also changed the content of the resulting texts, so it was not possible to reliably compare its success based on words. For this reason, we did not include its results in the overall results. Besides, in the case of SAS, we also performed experiments for the 6-gram algorithm as shown in Table IV and Fig. 6.

TABLE IV. RESULTS OF ALGORITHMS ON WIKIPEDIA DATASET SORTED BY SUCCESS RATE

Algorithm	Error rate	Success rate
Our algorithm (disk)	1.33%	98.67%
Our algorithm (memory)	1.53%	98.47%
BRM	1.70%	98.30%
SAS (6-grams)	2.38%	97.62%
SAS (4-grams)	2.47%	97.53%

We did not have the opportunity to carry out our tests with data using a neural network as described in [7], where the success rate for the Slovak language reached 98.84% and using the language model even 99.09%. This shows that the

neural network achieved the best results at present. Despite this, our proposed algorithm achieves better results compared to other algorithms freely available and accessible on the web.

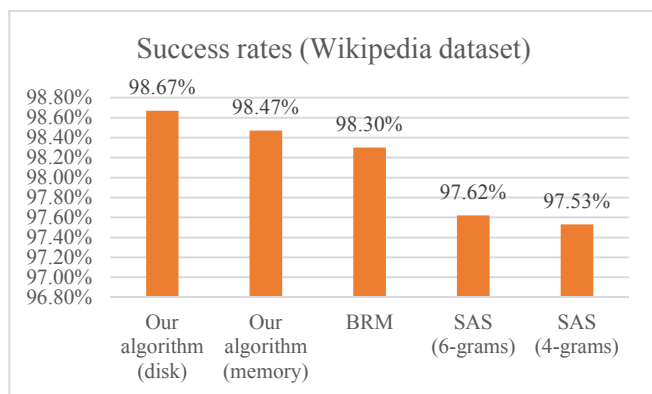


Fig. 6. The chart of overall success rates of the algorithms on Wikipedia dataset

A. Our algorithm in memory

The first algorithm with the data structure in the main memory reached a total success rate of 98.07%, using n-grams diacritics for 83.03% of words were reconstructed. The maximum size of data in main memory was 4.28 GB. The average speed of the reconstruction of one word was 0.24 ms. Table V shows the overall results.

TABLE V. TEST RESULTS – STATISTICS FOR OUR ALGORITHM IN MEMORY (OUR DATASET)

	Professional	Journalistic	Artistic	Total
Words in text	18437	17672	17187	53296
Use of 4-grams	373	576	198	1147
Use of 3-grams	2014	2884	1687	6585
Use of 2-grams	6475	7591	8027	22093
Use of 1-grams	5197	3917	5315	14429
Use of all n-grams	14059	14968	15227	44254
Time (ms)	3994	4363	4301	12658
Error words	25	223	513	1026
Success rate	99.86%	98.74%	97.02%	98.07%

B. Our algorithm in disk

The results of the second algorithm with data structure on the disk shown in Table VI. This algorithm has a better success rate than previous but is slower. It was used up to 44.56% of all n-grams from the cache. The maximum size of data in RAM was 1.16 GB and the average speed of the reconstruction of one word was 4.17 ms.

C. SAS

The E. Štúr Institute of Linguistics of the Slovak Academy of Sciences has 10 methods for reconstruction diacritics: first (selection of the first word), random (selection of random word), naive (selection of the most frequent word), n-gram (selection of word based on n-grams where n can be 2, 3, 4, 5 or 6), surreal and maximalist. The default method is 4-grams.

We found a few shortcomings of the reconstructor:

- Ignore multiple spaces – if there are more than one white space or white space between words, the SAS ignores them and use only one.
- Adding diacritics to the Slovak word “a” (in English: “and”) – the word “a” without diacritics is the most common in the Slovak language. Nevertheless, the reconstructor very often added the accent “à”, “á” or “â”.
- URLs – accents do not usually appear in URLs, but they are still added.

D. BRM

Although the BRM works only on adding the most likely word principle (or the highest occurrence), it achieves a very good success rate. It does not contain issues described in the previous text.

E. STUBA

Using our test texts and texts from Wikipedia we uncover some errors of this diacritics reconstructor:

- It ignores multiple spaces and in the result are replaced by one space.
- Upper and lower-case letters are not formatted as in input text. This makes a word as names and titles wrongly written. Such as MP3 -> mp3
- Issues with quotation marks - quotations marks can be presented with multiple symbols (“”, “”, ‘’, ‘’, ... :). So whole words were missing.
- Numbering – if a word contains a number, this number is replaced by various letters: пaлпeрт8 (18), яззызкь (33), в8тa1йнэ (81), озтe1ць (31), etc.
- Words “ty”, “TY”, “Ty” and “tY” throws an exception and make the application to crash.

TABLE VI. TEST RESULTS – STATISTICS FOR OUR ALGORITHM IN DISK (OUR DATASET)

	Professional	Journalistic	Artistic	Total
Words in text	18437	17672	17187	53296
Use of 4-grams	835	1476	559	2870
Use of 3-grams	3054	4538	3061	10653
Use of 2-grams	5338	5915	6091	17344
Use of 1-grams	4832	3039	5516	13387
Use of all n-grams	14059	14968	15227	44254
Words from cache	6402	6518	6801	19721
Words from cache (%)	45.54%	43.55%	44.66%	44.56%
Time (ms)	61708	79790	76311	217809
Error words	26	200	508	975
Success rate	99.86%	98.87%	97.04%	98.17%

VI. CONCLUSION

In this work, we dealt with the problem of diacritics reconstruction from texts without diacritics. We stated the reasons why such texts are created at all and what is the problem when adding diacritics. We described what algorithms exist for the Slovak language and propose our

algorithm for reconstruction of diacritics with two implementations. We tested the algorithms, collected the results and compared them. Compared to other available algorithms and services, our proposed algorithm has achieved the best results and the lowest error rate. Nevertheless, there are other algorithms based on neural networks that perform much better results.

In the future, we would like to focus on improving the algorithm using neural networks, namely bidirectional LSTM with attention blocks.

VII. ACKNOWLEDGMENT

This publication was realized with support of the Operational Program Research and Innovation in frame of the project: ICT for smart society, code ITMS2014 +: 313011T462, co-financed by the European Regional Development Fund.

VIII. REFERENCES

- [1] "Merriam-Webster dictionary," 2019. [Online]. Available: <https://www.merriam-webster.com/dictionary/diacritic>.
- [2] W. Jansen and A. Delaitre, *Mobile Forensic Reference Materials: A Methodology and Reification*, US Department of Commerce, National Institute of Standards and Technology, 2009.
- [3] S. A. o. S. Ľ. Štúr Institute of Linguistics, *Pravidlá slovenského pravopisu. 3., upravené a doplnené vyd.*, Bratislava: Veda, 2000.
- [4] D. Hládek, J. Staš and J. Juhár, "Diacritics Restoration in the Slovak Texts Using Hidden Markov Model," in *Human Language Technology. Challenges for Computer Science and Linguistics. LTC 2013. Lecture Notes in Computer Science*, Springer, Cham, 2013.
- [5] R. Krchnavy and M. Simko, "Sentiment analysis of social network posts in Slovak language," in *2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, Bratislava, 2017.
- [6] A. Hucko and P. Lacko, "Diacritics Restoration Using Deep Neural Networks," in *World Symposium on Digital Intelligence for Systems and Machines (DISA)*, Kosice, 2018.
- [7] J. Náplava, M. Straka, P. Straňák and J. Hajic, "Diacritics restoration using neural networks," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, 2018.
- [8] A. Novak and B. Sikósi, "Automatic Diacritics Restoration for Hungarian," in *2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.
- [9] D. Tufiş and A. Ceaşu, "DIAC+: A Professional Diacritics Recovering System," in *Proceedings of LREC 2008*, Marrakech, Morocco, 2008.
- [10] I. Zitouni, J. S. Sorensen and R. Sarikaya, "Maximum entropy based restoration of Arabic diacritics," in *ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006.
- [11] T. A. Luu and K. Zamamoto, "A Pointwise Approach for Vietnamese Diacritics Restoration," in *2012 International Conference on Asian Language Processing*, Hanoi, Vietnam, 2012.
- [12] "DIAKRITIK – nástroj na rekonštrukciu diakritiky (tool for diacritics reconstruction)," Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, [Online]. Available: <https://www.juls.savba.sk/diakritik.html>. [Accessed 2019].
- [13] J. Gedera, "Doplnovač diakritiky," Faculty of Informatics and Information Technologies, Slovak Technical University, 2015. [Online]. Available: <http://text.fiit.stuba.sk:8081/>. [Accessed 2019].
- [14] R. Hraška, "Doplnáč diakritiky," [Online]. Available: <https://diakritika.brm.sk/>. [Accessed 2019].
- [15] "Slovak National Corpus," Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, [Online]. Available: https://korpus.sk/index_en.html. [Accessed 2019].
- [16] M. Federico, N. Bertolti and M. Cettolo, "IRSTLM Toolkit," 2008. [Online]. Available: <https://hlt-mlt.fbk.eu/technologies/irstlm>.
- [17] J. Náplava, M. Straka, J. Hajič and P. Straňák, "Corpus for training and evaluating diacritics restoration systems," LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, [Online]. Available: <http://hdl.handle.net/11234/1-2607>.

Graphical Development Environment for Object Programming Teaching Support

Monika Václavková, Marek Kvet, Peter Sedláček
 Faculty of Management Science and Informatics
 University of Žilina
 Univerzitná 8215/1, 010 26 Žilina, Slovakia
 {monika.vaclavkova, marek.kvet, peter.sedlacek}@fri.uniza.sk

Abstract—The main goal of this contribution is to present the current solution of a graphical development environment for object programming teaching support. Presented software tool was created to simplify the understanding process of the basics of the object-oriented programming for JAVA language beginners. Therefore, we have been developing an application called FRIMAN at the Faculty of Management Science and Informatics at the University of Žilina. Suggested system consists of several cooperating modules. In this paper, we focus on description of selected two modules and their current functionality. The attention is paid to a module for class management and a graphical code editor, which enables creation of code bodies of methods by flow diagrams without the necessity of programming language syntax knowledge. Based on good evaluation of the development team preparing changes in the high school education process, the current application is planned to be applied in practice.

Keywords—Informatics teaching process, software tool, object programming, class management, graphical editor

I. INTRODUCTION

The very rapid development in the field of information technologies has become an important part of human lives. That is why many companies seek for young professionals, who are able to adapt to new or various changing trends very fast. One of the most affected subjects of these increasing changes are also the universities, which provide education of future professionals.

Graduates of any technically oriented university must get professional education in their particular study field. First, they need practical skills supported by excellent theoretical knowledge. The skills facilitate entry into employment while theoretical knowledge makes it easier to develop new unique solutions to problems that can arise in practice. Students of information technologies and management science are not an exception.

Faculty of Management Science and Informatics of the University of Žilina has been continuously making an effort to help their students mainly of the management study field to either develop or improve their understanding and knowledge of modern information technologies. The result of this effort is an ability of future managers to use the modern information technologies as a competition tool for creating a positive and efficient organizational change or bringing new solutions based on information technologies.

Nowadays, our current students understand information technologies and their background quite well. Mentioned information technologies are used by millions of users with the aim to maximize the efficiency of the world and society

development process. Despite the basic knowledge of various operating systems performance and common office applications, our students should learn also the basics of advanced information systems engineering [1].

Students of the management science at our faculty have to complete several subjects aimed at software development. Many of these students have either very weak or even no experience with any kind of programming. Therefore, it is very hard for them to understand the principles of different programming strategies like object-oriented programming [2, 3].

Object-oriented programming (OOP) belongs to the family of the most important and the most commonly used programming paradigms. It is based on the concept of so-called "objects", which can contain data, in the form of fields known as attributes, and code, in the form of procedures called methods. The methods are often used to modify the data fields of the object which they are associated with. For many beginners in software engineering, even the basic logic of a simple program based on OOP principles constitutes a challenging task, because many applications are designed by making them out of objects that interact with one another [3, 4, 5], what differs from the style of procedural programming [6], which some students are familiar with.

Another big issue may follow from the correct syntax, i.e. when, where and why to write a semicolon or any other specific symbol. Such problems often occur when changing current programming language for a different one. Therefore, we have been searching for a suitable way of making the education process of the programming basics easier mainly for the beginners.

The FRIMAN project originally introduced in [7] and reported in [8, 9] was designed and created as an answer to mentioned problems of the beginners in software developing. It is a useful software tool to support education of the object-oriented programming in Java language at our faculty. Our students have developed it for several years.

The main research topic refers to and the contribution of the paper is to be seen in the development of mentioned FRIMAN system and its particular components.

The remaining part of this paper is organized in the following way. The next section provides the readers with the basic characteristics of the FRIMAN project and it describes the architecture of the suggested system. The third section is devoted to the explanation of the Class viewer module and the fourth section summarizes the features of the Code editor module. Finally, the last section contains the obtained results and future development directions.

II. THE BASIC CHARACTERISTICS OF FRIMAN

The first aspect needed to be taken into account is the expected target group of users of the developed system and its requested primary functions. At the same time, it is sufficient to analyze existing software solutions in this area.

Currently, there are many environments, which offer the possibility of simple software projects development. The comparison of selected suitable solutions has led to the conclusion that the suggested system FRIMAN should find such ways of interaction, which would be easy to understand by various groups of users, mainly beginners in object-oriented programming. By this software tool we want to provide the users with a simple and interactive style of communication and control.

An example of suitable existing system could be the well-known environment BlueJ [10]. Currently, it is being used in our teaching courses for the beginners. In contrast with professional environment like IntelliJ [11] or NetBeans [12], BlueJ offers simpler interface for classes and their instances management via a graphical tool for object inspecting. This feature enables the users to thoroughly understand the basic principles of the object-oriented programming.

The main difference between the existing solutions and the suggested developed tool consists in usage of graphical components for creating the algorithms without the necessity of the knowledge of the specific programming language syntax. Instead of writing the code, various graphical components should be used by the program developers to create their application. The resulting code will be generated from the diagram itself afterwards. The suggested environment should check the correctness of the diagram and allow the users to debug their resulting algorithms.

Based on the mentioned primary features and other supplementary system requirements, we have suggested the following architecture of the FRIMAN software tool. The suggested system consists of five modules:

- Editor,
- Debugger,
- Builder,
- Class viewer,
- Core.

The first module is the Editor. It was developed for creation of individual methods and constructor bodies making use of common flowcharts, which can be made in an interactive way. This module will be explained in detail in the next section.

Algorithms made in the Editor module are automatically compiled to the JAVA language using the module Builder.

The next module is a Class viewer. Its main task is to provide creating classes and their management and also making instances of these classes. This module will be also described in a separate section.

The Debugger module checks possible logical errors in the proposed algorithms. It processes all used graphical components in the resulting flowchart and based on the

corresponding programming code it detects source of errors. This module is currently being implemented.

Finally, the Core module connects all satellite modules and provides proper performance of the whole application. Obviously, the FRIMAN software architecture was designed with regard to future system enlargements.

III. THE CLASS VIEWER MODULE

The Class Viewer module was suggested for managing classes and their instances. With respect to future users' abilities and experience, we are looking for the simplest and the most understandable way, in which the users could communicate with the application and create classes, their instances and processing various operations with them. We were inspired by the commonly known JAVA development environment BlueJ [10], which allows using the inspector function as a specific tool.

The basic functions, which should be implemented in our module for effective class management first, could be summarized as follows:

- Class management,
- Creating object instances,
- Inspecting inner values of attributes in instances,
- Calling methods and sending messages to particular instances and their ancestors,
- Passing the instances as parameters.

The use case diagram of required functionalities of the Class Viewer module is depicted in Fig. 2.

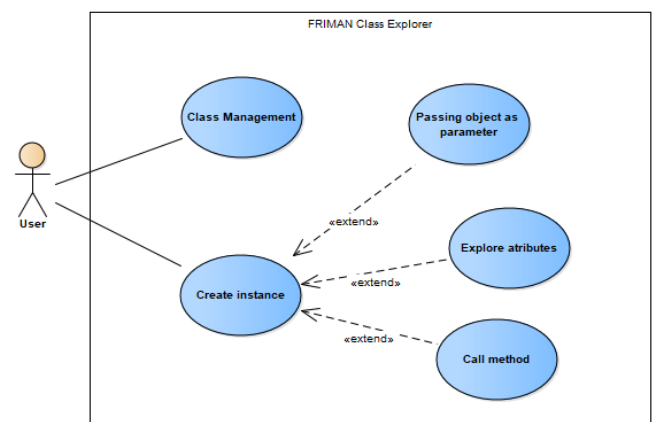


Fig. 1. A use case diagram for the Class Viewer module

Based on specified system requirements, the module was suggested and implemented in such a way that allows its users to get the possibility to manipulate with classes by their own requirements. It provides also functionality to create new instances, modify them and delete the created objects. The instances can be also used as parameters of messages sent to other instances or as other objects constructor parameters.

Since the FRIMAN software development tool has been implemented in JAVA programming language, one of the very important tools used in the Class Viewer module implementation was the Reflection API.

Reflection is a feature in the Java programming language that allows an executing Java application to examine or

"introspect" upon itself and manipulate internal properties of the program. It is possible to get the names of all class fields and to display them. The ability to examine and manipulate a Java class from within itself may not sound like very much, but in other programming languages this feature simply does not exist. For example, there is no way in a Pascal or C program to obtain information about the functions defined within that program [13, 14].

Reflection API is symmetric, that means that if we hold an object of a class, we can analyze its inner parts and contrary, if we have one of its inner parts, we are able to find out, which class declared this part. This way, we can move both directions from a class to a method, to a parameter of a class, etc. One of interesting applications of this technology is to find out the most of relative dependences between given class and the rest of the system[15].

The Class Viewer module also cooperates with a file system. All configuration files for classes created in a concrete project developed by our application are saved in suggested file system. The module creates class instances and these objects will be saved in operation memory. Instances enable us to call partial operations. The process can be formally described by the following activity diagram, which is depicted in Fig. 3.

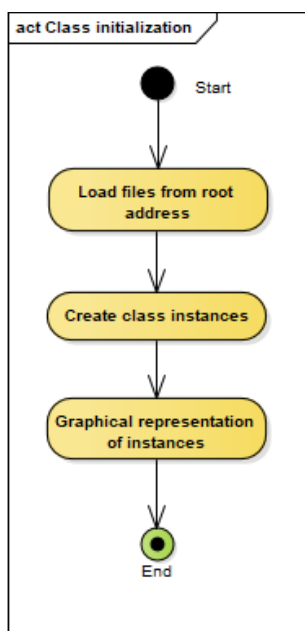


Fig. 2. An activity diagram of the class initialization process

The first step of the class initialization process is reading the necessary configuration files from particular folder of the associated file system.

Based on loaded configuration settings, the following step consists in creating the class instances.

Finally, all created objects - class instances are visualized in a specific graphical environment. Mentioned graphical environment for the class instances visualization will be explained in the next parts of this paper.

Inner object state can be inspected as well as particular object attributes of any instances. Furthermore, the module enables to call any method not only of the associated object

itself, but also any inherited method. In case, when a method return value is represented by an object, the system enables to create and save an instance of this object. This process is reported in Fig. 4.

The first step of the method execution process is calling the method itself. Thus, the code of the method is executed. If the execution of the method finishes by returning any value or an object, then the return value is displayed in a separate graphical element.

If the return value was an object, the module enables to inspect inner parts of this object. This functionality is similar to those, which many BlueJ users are familiar with. Another possibility consists in using the return value as a new object instance to a specific graphical environment for further processing.

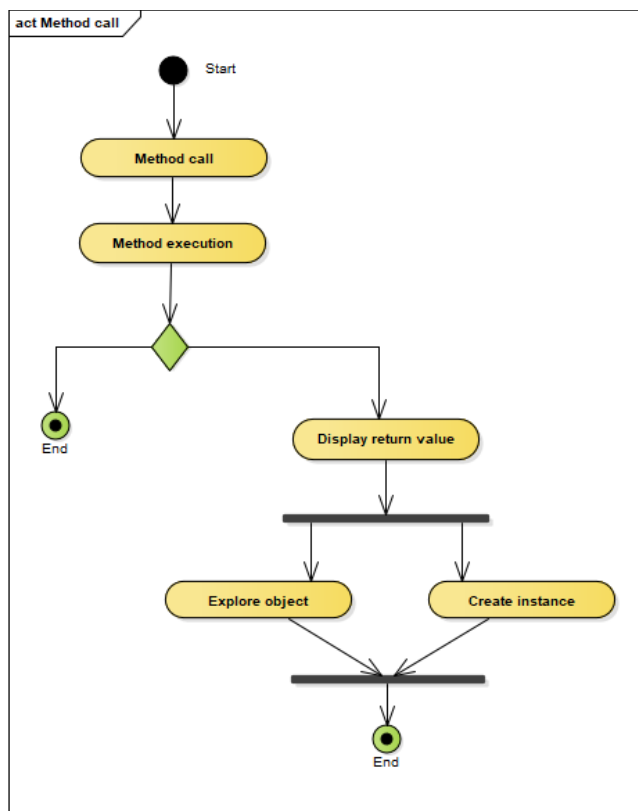


Fig. 3. An activity diagram of the method execution process

The following Fig. 4 depicts the main form of the resulting application in the Class Viewer mode. Obviously, the view mode can be easily changed to the Editor mode, which is reported in Section IV.

The top part of the depicted view contains the main menu of the application. The left part of the window is divided into two parts. The upper part enables the users to inspect the list of classes in the selected project. The users can choose any of them to be inspected. The lower part is used to display the definitions of all object attributes, methods and used imports.

The main part of the window is in right side. This section can be also divided into two parts. Detailed view can be seen in Fig. 5. The upper part of this figure visualizes all classes

with their mutual possible relations. Similarly to the BlueJ environment, our Class Viewer module enables to select any of classes by mouse click on particular class representation and create an instance of this class.

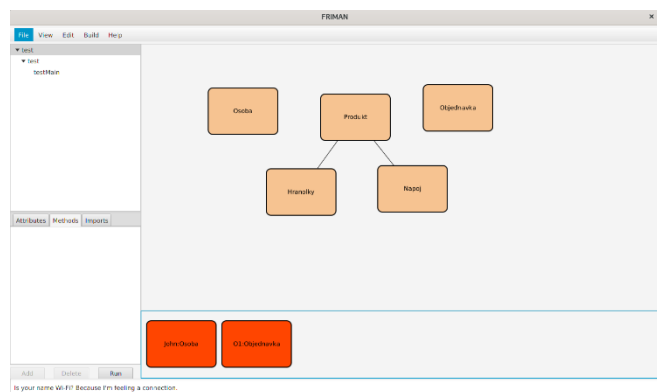


Fig. 4. The main form of theFRIMANsystem in theClass Viewer mode

All created instances of classes depicted in the upper part of the window, are placed in the bottom part of the window. The graphical representation is designed similarly to the common environments. Here, the users have the possibility to click on any object instance and then call any method of the selected object by sending message from its context menu.

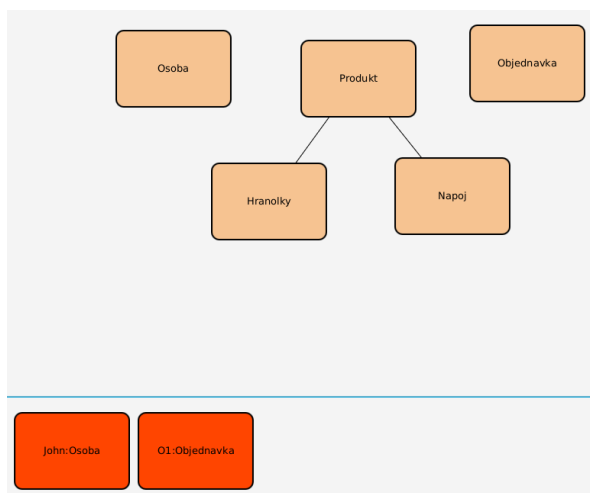


Fig. 5. Detail of Class viewer module

Except calling member methods, user is able to explore selected instance and view values of its fields, as can be seen in Fig. 6. This functionality is also available from context menu of selected instance.

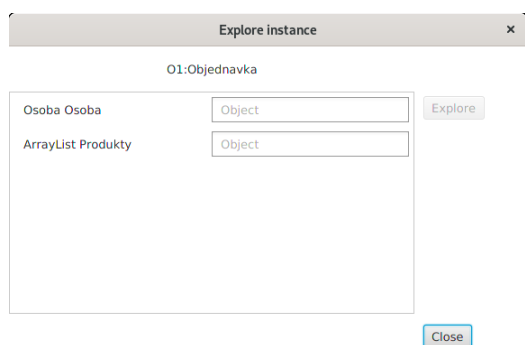


Fig. 6. Example of instance exploration

IV. THE EDITOR MODULE

The Editor module was suggested and incorporated to the FRIMAN application because of the necessity of creating and implementing methods and constructor bodies. Our vision was to use activity diagrams for this purpose and to make the process of making the algorithms interactive.

During the development of mentioned module, we have considered future system users, mainly students, beginners. The interactive way of programming is assumed to simplify the learning process of the object programming paradigm and the basics of creating algorithms at the same time. We believe that they will be able to understand the process of creating algorithms inside object methods using this simpler form without the necessity of programming language syntax knowledge.

The main functions that are required in this module can be divided into the following tasks:

- Loading the library of graphical components, which are necessary for creating activity diagrams.
- Loading and displaying activity diagrams from a file.
- Creating the activity diagram using offered graphical components.
- Possibility of editing existing activity diagrams.
- Modification of text parts and scaling the graphical components.
- Generating the source code from the activity diagram to the JAVA programming language.
- Saving the final version of the activity diagram into a file.

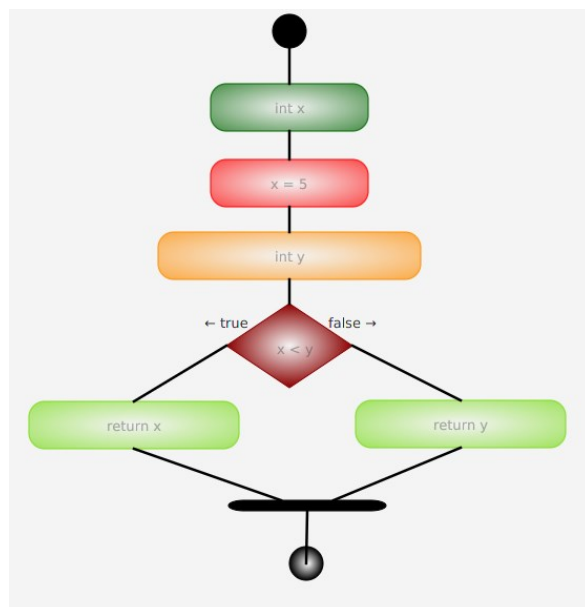


Fig. 7. Example of an activity diagram in the Editor module of FRIMAN

Based on the requirements summarized above, the Editor module has been suggested and then implemented in such a way that it enables intuitive and plain mastering by its users. The module allows creating activity diagrams making use of the set of pre-defined graphical components. Example of this

diagram can be seen in Fig. 7. These graphical elements are modifiable by the needs of users. The provided set of graphical elements offers all basic programming schemes (input, output, assignment, branching, cycles, etc.). Available graphical elements can be seen in Fig. 8.

It is also possible to edit the text information inside the graphical components as well as scale the component itself according to the user's concepts and requests.

One activity diagram can consist of many different graphical components, which are joined by connections. The connection provides proper traceability between individual parts of the activity diagram. Any graphical component may be used multiple times according to the created algorithm. Furthermore, they can be hierarchically inserted into each other. If the connection between two parts is no longer needed, the module enables functionality to remove it.

During the process of drawing the diagram, the user is offered to switch to such a mode, in which the source code in JAVA programming language reflecting the activity diagram is being generated (Fig. 9). Mainly the advanced FRIMAN system users will appreciate this functionality. Another functionality allows users to switch between more activity diagrams opened at the same time in various tabs.

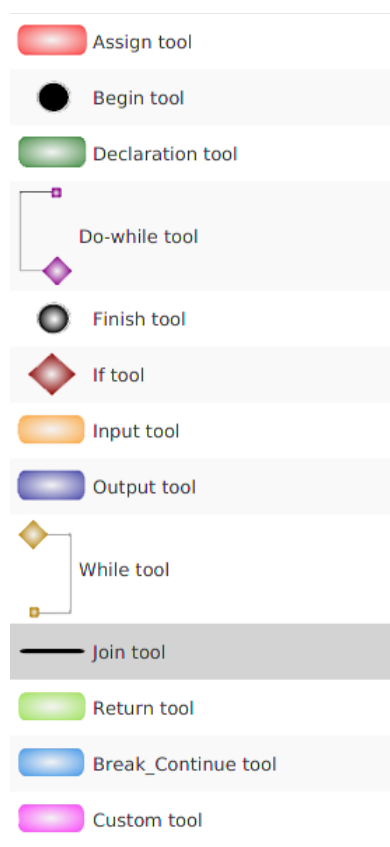


Fig. 8. Toolbar for Activity diagram

The main window of the Editor module in the application is similar to Class Viewer, except right panel which is in this case used to display tabs with opened activity diagrams and generated source code.

The most important part of the Editor module is depicted on the middle and right side of the main window and it is divided into two parts.

The right part contains the pre-defined set of offered graphical components, which can be used for creating the activity diagram.

The biggest part of the main window is placed in the middle part and it represents the concrete activity diagram made from components offered in the right side.

The diagram is created using the drag-and-drop system. The user chooses any component from the right panel and uses a mouse click to insert the component into the diagram. Then, the text fields of selected component are edited. In case, when two or more components are needed to be connected, the user chooses a join component to make the relation between the components.

The set of created graphical components enables to create any algorithm. The components have been chosen in such a way that the basics of any programming language are covered by this set. Thus, this functionality enables future extension of the existing system by the possibility to translate the activity diagram to different programming languages. The current version supports only the JAVA language.

When a user switches the code tab, the activity diagram is transformed into the source code in JAVA language. Simultaneously, the generated code is compiled by a JAVA compiler, which is a part of another module – Builder, which is currently being developed.

If there are any mistakes or errors, they are detected. The user should make corrections in an interactive way, e.g. in the graphical mode. In other words, it is not necessary to edit the programming source code in JAVA. When there are any changes performed in the activity diagram, the source code is generated and compiled again. This process can be repeated until obtaining the right result.

The example of a generated source code is depicted in Fig. 9.

```

{
    int x;
    x = 5;
    Scanner scanner = new Scanner(System.in);
    int y = scanner.nextInt();
    if(x < y){
        return x;
    }
    else {
        return y;
    }
}

```

Fig. 9. Example of a source code generated from the activity diagram

A. Source code generating

The main issue editor module has to solve is how activity diagram should be transformed into source code. There are many ways how this can be handled. In our case we chose the simplest one. Each element that can be used represents specific statement of code. For example, while element is generated in following way: firstly, keyword while is generated that is followed by logic condition. As next all

connected elements are generated to the end point of while element (yellow square in Fig. 8). All this is performed with language specific format, i.e. where language expects round brackets, round brackets are generated etc. Similarly, each element has its own way how to be generated. In other words, each element is responsible for its own generating into source code. Except that, each element has reference to element connected to it.

Thanks to this responsibilities distribution, we can start generating our source code from begin tool. After begin tool is generated, generating for its followers are called. This process continues to the point, where whole method is generated. Please note, that this transformation is possible only in case, activity diagram does not contain any cyclic connected elements. Therefore some constraints for connection has to be implemented. In our tool we decide, each element can have exactly one parent element, i.e. element cannot have more than 1 ingoing connection. Begin tool cannot have any ingoing connection and finish element cannot have any outgoing connection. Source code is always generated only from begin tool. Thanks to these facts, it is not possible to cyclic connect elements and our transformation is working.

The described process is only used for generating source code of method. In order to run compiler and build whole project also other parts has to be generated. For these purposes we decide to implement some business objects, that will hold information about various parts of project, user is working on. Please note, that this is no longer responsibility of editor module, but explanation of this is also required to understand whole process of generating source code. These business objects include package descriptors, class descriptors etc. Package descriptor is generated as folder in file system. Each package contains references to its classes. Class descriptor contains full information about one class, i.e. access modifier, implementing interfaces, fields etc. All of this information is also generated into corresponding file with exact syntax of java language. Class descriptor also contains information about its method descriptors. Method descriptor holds information as method access modifier, return type etc. Except that method descriptor holds also its activity diagram. When generating, method descriptor generates firstly its head and then calls generating of source code from activity diagram. This is also included when class is generating itself. This leads to challenge – as one class holds many methods, how this process of generating can be optimized so as not to whole file to be generating each time one element in one method is changed.

V. CONCLUSIONS

This paper was focused on the application called FRIMAN developed at the Faculty of Management Science and Informatics at the University of Žilina. The system consists of several cooperating modules. In this paper, the current versions of the Class Viewer and the Editor modules were presented.

During the development process, many different challenges for future work on the system have occurred. First, we need to finalize the implementation of the missing modules. Then, the cooperation among modules may need to be resolved. The biggest challenge from our point of view is the Debugger module, because it should be done also in a graphical way. Another research topic that needs to be solved in the future work consists in optimization of code generating as mentioned in previous sections.

Based on reported results we can conclude that the FRIMAN system represents a very useful tool for beginners in object programming subjects.

ACKNOWLEDGMENT

This paper was supported by the grant system of the University of Žilina.

REFERENCES

- [1] D. E. Knuth, *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1997.
- [2] B. Stroustrup, "What is object-oriented programming?," *IEEE Software*, vol. 5, no. 3, pp. 10–20, May 1988.
- [3] O.-J. Dahl, "The birth of object orientation: the Simula languages," in *From Object-Oriented to Formal Methods: Essays in Memory of Ole-Johan Dahl*, O. Owe, S. Krogdahl, and T. Lyche, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 15–25.
- [4] A. Schreiner, *Object-Oriented Programming With ANSI-C*. Axel T. Schreiner / Lulu, 2011.
- [5] E. Kindler, I. Krivy, "Object-Oriented Simulation of systems with sophisticated control," *International Journal of General Systems*, vol. 40, no. 3, 2011, pp. 313-343
- [6] H. Jordan, G. Botterweck, J. Noll, A. Butterfield, and R. Collier, "A feature model of actor, agent, functional, object, and procedural programming languages," *Science of Computer Programming*, vol. 98, pp. 120–139, Feb. 2015.
- [7] E. Paršo, et al., "FRIMAN," *Central European Researchers Journal*, vol. 2, no. 2, 2016, pp. 70-76.
- [8] P. Sedláček, M. Václavková, "Tool for supporting Education process in Information technology," *ICETA 2018: 16th IEEE International Conference on Emerging eLearning Technologies and Applications*, 2018, pp. 483-488.
- [9] J. Kostolný, M. Václavková, "Learning system FRIMAN," *ICTERI 2017: proceedings of the 13th international conference on ICT in education, research and industrial applications*, 2017, pp. 281-288.
- [10] "BlueJ". [Online]. Available: <https://www.bluej.org/>. [Accessed: 22-Oct-2019]
- [11] "IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains," JetBrains. [Online]. Available: <https://www.jetbrains.com/idea/>. [Accessed: 22-Oct-2019].
- [12] "Apache NetBeans," Welcome to NetBeans. [Online]. Available: <https://netbeans.org/>. [Accessed: 22-Oct-2019].
- [13] Using Java Reflection. [Online]. Available: <https://www.oracle.com/technetwork/articles/java/javareflection-1536171.html>. [Accessed: 22-Oct-2019].
- [14] J. Jenkov, "Java Reflection Tutorial," Jenkov.com. [Online]. Available: <http://tutorials.jenkov.com/java-reflection/index.html>. [Accessed: 22-Oct-2019].
- [15] C. Momanis and C. Momanis, "Take an in-depth look at the Java Reflection API," *JavaWorld*, 01-Sep-1997. [Online]. Available: <https://www.javaworld.com/article/2077015/take-an-in-depth-look-at-the-java-reflection-api.html>. [Accessed: 22-Oct-2019].

Obtaining tire tread model from its real world photo

1st Michal Vagač

Department of Computer Science
Fac. of Sciences, Matej Bel University
Banská Bystrica, Slovakia
michal.vagac@umb.sk

2nd Michal Povinský

Department of Computer Science
Fac. of Sciences, Matej Bel University
Banská Bystrica, Slovakia
michal.povinsky@umb.sk

3rd Miroslav Melicherčík

Department of Computer Science
Fac. of Sciences, Matej Bel University
Banská Bystrica, Slovakia
miroslav.melichercik@umb.sk

Abstract—Tire prints are among the most common types of evidence at crime scenes. Therefore they represent important evidence, that can be essential for a culprit reveal. The important characteristics of a tire tread is its wearing down. This process may affect basic shape of tire tread print.

In the paper we present experiment, in which we use convolutional neural network to get 3D model of tire tread from its 2D photography. The model may be later used to simulate wear down of tire tread and thus to estimate new shape of its print.

Index Terms—forensic, traceology, tire tread print, image processing, convolutional neural network, deep learning

I. INTRODUCTION

Tire prints are among the most common types of evidence at crime scenes (in field of traceology it is the second most common, right after shoe prints – Fig. 1) [1].

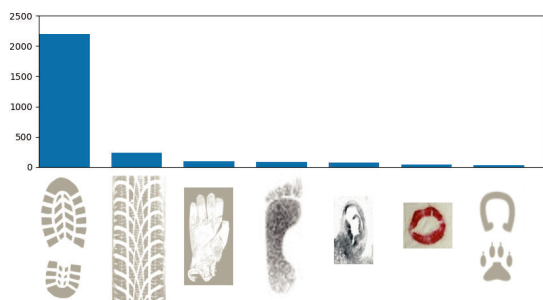


Fig. 1. Occurrence of traceology evidences in Slovakia during a year.

After securing a tire tread print at a crime scene, it is important to determine the brand or manufacturer of the tire. Such knowledge allows investigators to focus on a narrow group of tires. Determining brand of a tire can be achieved by using a database of tire tread samples and related parameters as name, manufacturer, etc. Examined tire print is then matched against this database. Complication occurs, when the inquired tire is worn down significantly. A print made with such a tire can differ from tread sample stored in the database, and so it would not be matched. For this reason, quality of database would be improved, when besides samples of treads of new tires, it would contain also samples for tires with different levels of wear down.

This information could be acquired from 3D model of tire treads. A tire wear down is in fact gradual reduction of the

topmost layer of a tread. This process could be simulated on 3D model using spatial transformations. Thus to realize this idea, 3D model of a tire tread is essential starting point. Since we aim to be able to identify an unknown tire print among database of known tire tread samples, we need as much tire tread models as possible. Manual creation of 3D models is time demanding task. Building a database with eventually thousands of samples would be unreachable goal.

On the other side, the Internet contains huge amount of photos of tires (in tire e-shops). Automatic creation of 3D model of tire tread based on its 2D photography would help to solve the problem of building database of 3D models of tire treads.

II. RELATED WORK

Number of research publications related to analysis of tire tread images is rather limited. The most papers focus on tire tread pattern image classification. Of those, majority focus on texture features as representation of tire tread pattern image.

In [2], tire tread images are pre-processed and normalized using 2D Fast Fourier Transform. Complexity of high dimensional image data is reduced using principal component analysis (PCA) and power spectrum analysis (PSA). The k-Nearest Neighbors (KNN) classifier is used to find the best matches in the database. This process allows very simple classification of the tire images. [3] proposes a tire matching system based on Gabor wavelets and support-vector machine (SVM) classifier. In the preprocessing step, the input images are processed by morphological opening to eliminate some fine holes in the image. The Gabor wavelets are then used to represent the features of tire tread patterns with different scales and orientations. The matching process is carried out by SVM classifier, Euclidean distance (ED) and cosine distance (CD). Paper [4] proposes a new method to calculate the contrast feature by making use of the statistical moments of intensity histogram to extract more information from the image for better description. An improved SIFT (Scale-Invariant Feature Transform) [5] feature point extraction method is presented in [6]. The improvement is achieved by removing low frequencies from the images, what significantly reduces the number of SIFT feature points (and thus improving the retrieval speed). Another approach [7] improves histogram of oriented gradients (HOG) in two aspects: at first it improves the representation precision of the extracted HOG feature, at second in introduces

the rotation invariance property. The proposed algorithm is evaluated on tire tread dataset.

Authors of [8] focus on problem of deviation between tread pattern and design pattern in the process of tire vulcanization. They propose a new method to extract and reconstruct tread pattern feature from tire tread scan data. In the first step, the pattern pitch arrangement is used to determine segmentation. Then, the contour of the tire crest is obtained. And at last, the point cloud is transformed into two-dimensional grayscale image. According to the growth rhythm of the pattern, its border characteristics is identified and extracted. The pattern boundary is discretized using cubic boundary spline interpolation method (into a series of equidistant coordinate points).

Paper [9] proposes tread pattern classification algorithm with feature fusion based on transfer learning. In the first step of the algorithm, knowledge of a pre-trained convolutional neural network (CNN) model on ImageNet dataset is transferred to a new model. The parameters of the new model are fine tuned using tread pattern image data. In the second step of the algorithm, features from multiple fully-connected layers are extracted. These represents high-level features of tread pattern images. In the third step, histogram of oriented gradients (HOG) is calculated as the low-level feature of the tread pattern image. In the fourth step, the features from the CNN model are combined with HOG as fusion feature. Finally, the fused features are used to train SVM classifier for tread pattern image classification. The concept of transfer learning solves the problem of lacking large training dataset, what is common problem in this research area.

III. IMPLEMENTATION

A. Dataset

The aim is to have automatic process for building a 3D model in such way, that its rendering would correspond to existing 2D photography. The process is simplified by fact, that we will be handling only specific kind of images (photos of tire treads). Since the task is relatively easily doable after short training by human, we suppose that it would be solvable also by properly trained neural network.

To train such a neural network we need a dataset containing 2D photographs and corresponding 3D models. However, it is very difficult to create a 3D model in such a way, that its rendering would result into existing 2D photo. Correspondence between the two must be very precise. Thus another possibility for dataset preparation will be in using result of 3D model rendering as alternative to 2D photography.

To model and render 3D scene we have used open-source 3D computer graphics software named Blender [10]. Its big advantage, besides its overall high quality and that it is free, is its ability to control all of its functions by Python scripts. This allows to automate rendering process for different camera and lighting settings, and thus to create large amount of renders easily.

As our first approach for building a dataset, we have used 3D models of whole tires. The 3D models were obtained from [11]. We have selected 7 different models of tires with

treads in detail. For each model, we have rendered images for about hundred different camera settings. Each rendering resulted in pair of rendered image and corresponding depth map (an image containing information relating to the distance of the tire tread from camera [12]). Example of such pair is depicted in Fig. 2.



Fig. 2. Rendered image (left) and corresponding depth map (right).

Resulting dataset was used to train convolutional neural network described in III-B. However, ability of network to work with real world photos was very limited. One of the reasons was curvature of tire, which caused extensive changes in depth map (compared with the tread depth).

The described shortcoming led us to use a different approach. Instead of model of whole tire we have decided to focus on tire tread modeled on a plane. Modeling of bumpy surfaces can be achieved by technique named displacement mapping [13]. This approach uses height map to displace geometric positions of points over the textured surface. Amount of actual displacement is proportional to value (intensity) in the height map (image).

This approach was even easier to realize. Modeling of tire treads consisted of preparation of 2D grayscale images, in which intensity of each pixel represented amount of displacement (Fig. 3).

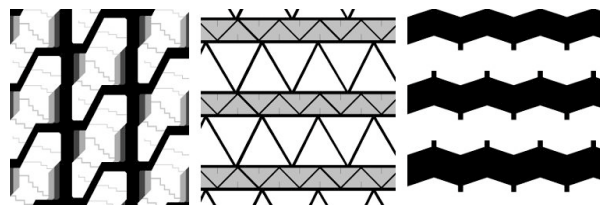


Fig. 3. Example of images used as displacement maps.

Prepared images were used as displacement maps and together with a material of rubber were assigned to a plane. This simple scene was used to render images for training dataset. For each one displacement map a set of rendering results was obtained as a result of following steps:

- Smooth camera movements within specified range provided different views on the scene.
- For each camera movement:
 - Color of rubber material changed randomly (within specified range).
 - Light position changed randomly (within specified range).

For each one displacement map we have set one hundred different camera positions. Considering few extra renders for

color and light changes, we have got few hundreds images for each displacement map.

As a rendering result, we have again expected a pair – realistic looking image of tire tread and corresponding image of depths in each point in the image. The first image we have got as a render of scene using rubber material. The depth information was effectively stored in the images used as displacement maps. However, since the scene was rendered from different camera views, this fact has to be included in depth image. Instead of using depth map, we have assigned displacement map as a material texture.

Thus each scene configuration was rendered twice – once with a rubber material (producing realistic view of scene), and second with the displacement map used as a material (producing image with same geometry of the scene, but with intensities describing depth of the tread). The important point was to disable shading completely, so that intensities were not affected by lighting conditions. Example of the couple including used displacement map is depicted in Fig. 4.

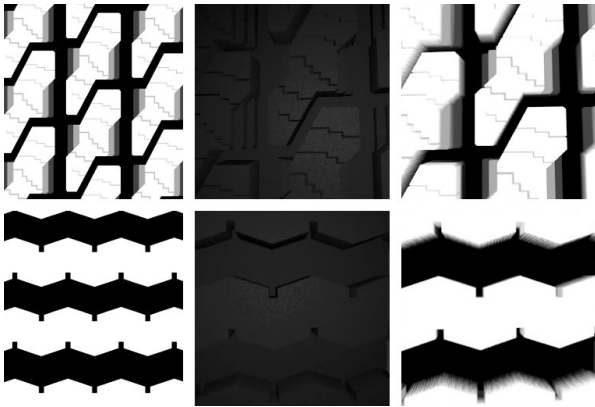


Fig. 4. Example of data from dataset – displacement map (left), image rendered with a rubber material (center) and image rendered with displacement map as a material (right).

As the last step we have extended the dataset with real world photographs of tire treads together with manually labeled tread depth at each point of the image (Fig. 5).

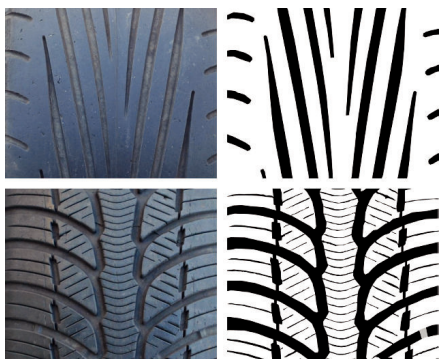


Fig. 5. Dataset was extended with manually labeled (right) real world photography (left).

B. Network

The neural network was implemented in Torch framework [14] using Lua language. It's architecture (Fig. 6) stems from U-Net convolutional neural network [15] and consists of 3x3 convolution layers with ELU (Exponential Linear Unit) activation function.

The numbers of layers, channels and nested modules was determined empirically to achieve acceptable output quality while keeping the computational costs low. We have evaluated different sizes of convolution kernels, different dropout values and non-linearities, and different number of nested layers. We have compared results of a standard convolution with results of MixConv [16]. In the final architecture are all convolutions padded to preserve image size. Every convolution layer except the final convolution is immediately followed by ELU activation function. The final convolution layer is followed by sigmoid activation. The network is trained using binary cross-entropy loss. It has 937945 parameters.

IV. EXPERIMENTS AND RESULTS

Using the procedure described in the implementation section, we have prepared dataset containing 1200 pairs of rendered images and corresponding depths, which we have supplemented with 27 pairs of labeled real world photographs. The rendered images were obtained from scenes with 6 different tire tread models.

During the network training, the dataset images were randomly cropped to size 200x200 pixels. For each few epochs, the network performance was evaluated by computing mean squared error (MSE) between the network result and ground truth data. Development of the MSE with respect to the epochs is presented in Fig. 7. After training for 212 epochs, the network was tested with real world photos of tire treads (not anyhow involved in the training process). Fig. 8 depicts two real world photos together with results obtained from the neural network.

As can be seen on the results, reconstruction of 3D model from its 2D image works. However, several defects can be seen especially in lower levels.

V. CONCLUSION

In this paper we have presented our approach of using convolutional neural network for building 3D model of tire tread from its 2D real world photography. Most of the work was devoted to dataset preparation. The best results we have achieved with dataset built using displacement maps. Each scene with 3D model of tire tread was rendered twice – once with rubber material to provide a real looking image, second time with displacement map as a texture to provide image with tread depths.

We suppose that the results can be further improved – by increasing the number of the samples in the dataset (both – artificially rendered and real world labeled). One disadvantage of tread depth description by displacement map is its inability to describe certain places in the scene. In the future we plan to

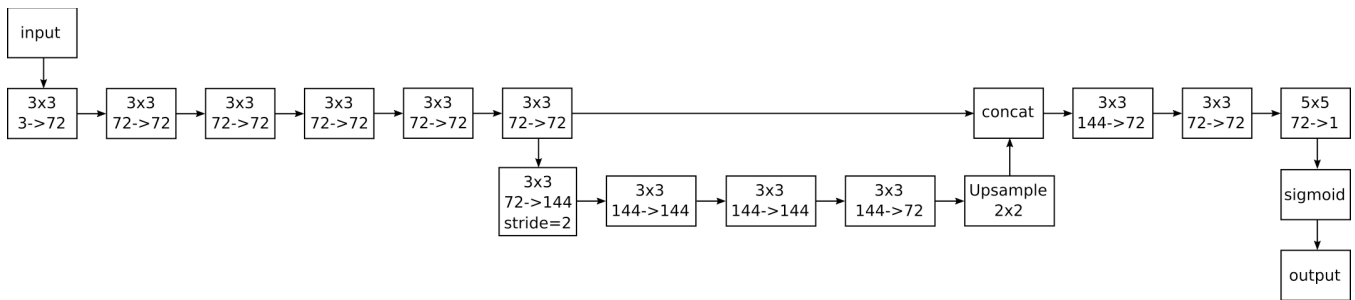


Fig. 6. Proposed network architecture.

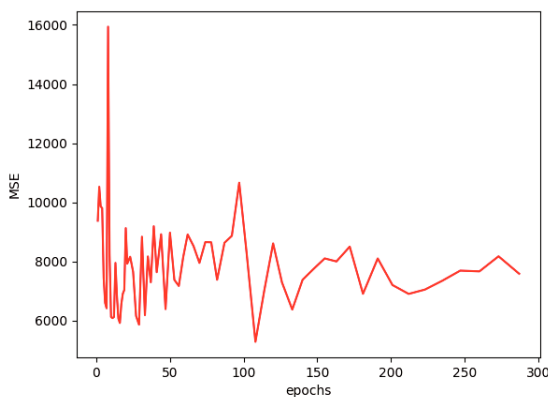


Fig. 7. Development of MSE with respect to training epochs

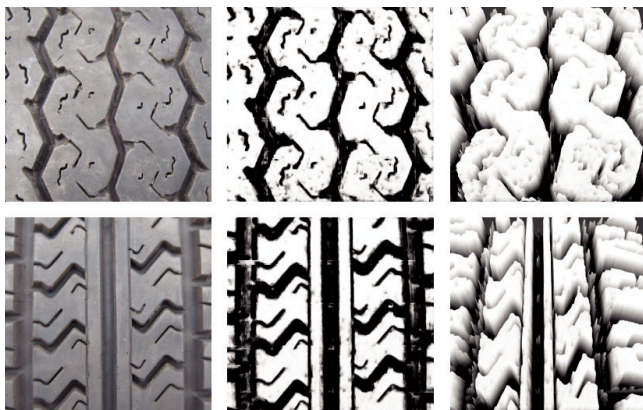


Fig. 8. Automatically created 3D model of tire tread (right) from its 2D photo (left).

focus on this problem, which solution would improve results even further.

ACKNOWLEDGMENT

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0219-12.

Computing was performed in the High Performance Computing Center of the Matej Bel University in Banska Bystrica

using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

REFERENCES

- [1] Institute of Forensic Science, Slovak Police Force, “Annual report 2014,” 2014.
- [2] D. Colbry, D. Cherba, J. Luchini, “Pattern Recognition for Classification and Matching of Car Tires,” *Tire Science and Technology* 33, 2–17, 2005.
- [3] D.-Y. Huang, W.-C. Hu, Y.-W. Wang, C.-I. Chen, C.-H. Cheng, “Recognition of Tire Tread Patterns Based on Gabor Wavelets and Support Vector Machine,” In: J.-S. Pan, S.-M. Chen, N. T. Nguyen (eds.), *ICCCI 2010, Part III, LNCS*, vol. 6423, pp. 92–101, Springer, 2010.
- [4] Y. Liu, Z. Li, Z. M. Gao, “An improved texture feature extraction method for tyre tread patterns,” *Intelligence Science and Big Data Engineering*, pp. 705–713, 2013.
- [5] D. G. Lowe, “Object recognition from local scale-invariant features”, In: *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, 1999.
- [6] S. Wang, Y. Liu, D. Li, H. Yan, B. Bai, “An Improved SIFT Feature Extraction Method for Tyre Tread Patterns Retrieval”, In: *Proceedings of the 2014 Seventh International Symposium on Computational Intelligence and Design*, pp. 539–543, December 13-14, 2014.
- [7] Y. Liu, Y. X. Ge, F. P. Wang, et al, “Texture feature extraction based on histogram of oriented gradient domain texture tendency for tyre pattern retrieval,” *IEEE International Conference Intelligent Control and Information Processing*, 2017.
- [8] Y. Dong, R. Zhang, Z. Song, Y. Fang, S. Bai, F. Zhang, “Research on Boundary Features Extraction and Reconstruction for Tire Tread Patterns,” *Journal of Mechanical Engineering*, vol. 54, no. 03, pp. 137–147, ISSN:0577-6686, February 2018.
- [9] Y. Liu, S. Zhang, D. X. Li, J. L. Fan, W. Liu, “An Effective Tread Pattern Image Classification Algorithm based on Transfer Learning,” In: *Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing (ICMSSP '18)*, ACM, New York, NY, USA, pp. 51–55, 2018.
- [10] <https://www.blender.org/>
- [11] <https://free3d.com/3d-models/>
- [12] https://en.wikipedia.org/wiki/Depth_map
- [13] https://en.wikipedia.org/wiki/Displacement_mapping
- [14] R. Collobert, K. Kavukcuoglu, C. Farabet, “Torch7: A Matlab-like Environment for Machine Learning,” In: *BigLearn, NIPS Workshop*, 2011.
- [15] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, Vol.9351: 234–241, 2015.
- [16] M. Tan, Q. V. Le, “MixConv: Mixed Depthwise Convolutional Kernels,” In: *Proceedings of the 30th British Machine Vision Conference (BMVC)*, 2019.

Modified Radial Basis Functions Approximation Respecting Data Local Features

Jakub Vasta

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
vastja@students.zcu.cz*

Vaclav Skala

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
skala@kiv.zcu.cz*

Michal Smolik

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
smolik@kiv.zcu.cz*

Martin Cervenka

*Department of Computer Science and Engineering
Faculty of Applied Sciences, University of West Bohemia
Plzen, Czech Republic
cervemar@kiv.zcu.cz*

Abstract—This paper presents new approaches for Radial basis function (RBF) approximation of 2D height data. The proposed approaches respect local properties of the input data, i.e. stationary points, inflection points, the curvature and other important features of the data. Positions of radial basis functions for RBF approximation are selected according to these features, as the placement of radial basis functions has significant impacts on the final approximation error. The proposed approaches were tested on several data sets. The tests proved significantly better approximation results than the standard RBF approximation with the random distribution of placements of radial basis functions.

Index Terms—Radial basis function, approximation, inflection points, stationary points, Canny edge detector, curvature

I. INTRODUCTION

The approximation is commonly used and well known technique in many computer science disciplines. This technique can be divided into two groups. The first one is the approximation that use the mesh and its connectivity. Some well known approaches that use the triangulation are [1]–[4]. However, all those approaches need the mesh connectivity, i.e. triangulation, which can be time consuming and difficult to compute for higher dimensions. On the opposite site, the second group are approximation techniques that does not require any mesh, i.e. they are called meshless methods. This paper focuses on this kind of approximation.

In the book [5] is provided an introduction for each of the most important and classic meshless methods along with the complete mathematical formulations. In total, it presents 19 meshless methods in detail with full mathematical formulations showing numerical properties such as convergence, consistency and stability. One example of approximation technique is Kriging [6]. It depends on expressing spatial variation

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, for their discussions and suggestions. The research was supported by the projects: Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

of the property in terms of the variogram, and it minimizes the prediction errors which are themselves estimated. Extension and variations of this method are available in [7]–[9]. Another approximation technique is weighted least square method [10], [11] it is simple because it is based on the well-known standard least squares theory. It is attractive because it allows one to directly use the existing body of knowledge of the least squares theory and it is flexible because it can be used to a broad field of applications in the error-invariable models. Very similar approach is the LOWESS method [12] which is used for meshless smoothing and approximation of noisy data.

The Radial basis function (RBF) methods have been widely used for approximation of scattered data, recently. A brief introduction to this method is in [13]. Comparison of different radial basis functions is in [14], [15]. The paper [16] presents an approach for large scattered data interpolation. It uses the space subdivision to reduce the computation time and more importantly to reduce the needed memory for RBF approximation. A modification of this algorithm for 3D vector field data approximation is presented in [17]. Very important for the final RBF approximation quality is the distribution of radial basis functions. This problem is described and suggested solution in [18], [19]. Many papers also propose a solution to the selection of the best shape parameters of radial basis functions [20]–[24].

II. RADIAL BASIS FUNCTIONS

The Radial basis functions (RBF) are commonly used for n-dimensional scattered data approximation and interpolation. This approach is used in many areas, e.g. image reconstruction [25], [26], neural networks [27] and surface reconstruction [28], [29]. The task can be stated as follow. Find analytic function for given pairs of values (x_i, h_i) , where x_i is a point position in n-dimensional space and h_i is value in this point. For such data it is not possible to use standard approximation and interpolation techniques because lack of

knowledge about data connectivity and ordering. Therefore, the RBF approximation has the following attributes:

- Designed for scattered data approximation/interpolation
- Independent of the data dimension
- Not separable, i.e. it is not valid to approximate/interpolate data "dimension by dimension"
- Invariant with respect to euclidean transformations

Hardy [30] proposed RBF interpolation based on interpolation equation:

$$f(\vec{x}) = \sum_{i=1}^M \lambda_i \theta(\|\vec{x} - \vec{x}_i\|), \quad (1)$$

where \vec{x}_i is data point and λ_i is point weight. $\theta(r_{ij})$ is radial basis function, where $r_{ij} = \|\vec{x} - \vec{x}_i\|$. Radial basis function can differ, but they can be divided into two main groups by their range of influence, i.e. global and local functions.

RBF approximation/interpolation leads to the linear equation system $A\vec{x} = \vec{b}$, where approximation differ from interpolation only with form of matrix A . Solvability and stability problems were solved for example in [31] [32]. Wright [32] extend original RBF interpolation with polynomial and added more conditions.

A. Radial Basis Functions approximation

RBF approximation is based on point distance in n-dimensional space and is derived from the same equation (2) as interpolation is.

$$f(\vec{x}) = \sum_{i=1}^M \lambda_i \theta(\|\vec{x} - \epsilon_i\|), \quad (2)$$

where M is number of radial basis functions, λ_i is weight of radial basis function, θ is radial basis function and ϵ_i is placement of radial basis function.

Given set of value pairs $\{\vec{x}_i, h_i\}_1^N$, where \vec{x}_i is point position in n-dimensional space, h_i is value in this point, N is number of given points. $N \ll M$ therefore we obtain over-determined system of linear equations.

$$h_i = f(\vec{x}_i) = \sum_{j=1}^M \lambda_j \theta(\|\vec{x}_i - \epsilon_j\|) \quad i = \{1, \dots, N\}, M \ll N \quad (3)$$

It can be rewritten in matrix form

$$A\vec{\lambda} = \vec{h} \quad (4)$$

This over-determined system of linear equations can be solved by LSE or QR decomposition.

III. PROPOSED APPROACH

Radial basis function placement is important factor for approximation error. In this contribution, property of these good points are proposed with the way to find them. First group are extreme points i.e. local/global minimum or maximum. Next group are points of inflection. These points represents changes in data. Another proposed group of points are stationary points of curvature. These points represents extreme curvature values and in some case they are similar to points of inflection. Last group are edge points as known from image processing, because we can look at data as on image depending on data structure or sampling. Search for important points is amended with Halton sequence [13] sampling with special stress on border and corner sampling for covering whole data set. The last step is reduction of points number with nearest neighbour condition.

The Halton sequence is computed using the following formula:

$$Halton(p)_k = \sum_{i=0}^{\lfloor \log_p k \rfloor} \frac{1}{p^{i+1}} \left(\left\lfloor \frac{k}{p^i} \right\rfloor \bmod p \right), \quad (5)$$

where p is a prime number, k is the order of the element of the Halton sequence, i.e. $k \in \{1, \dots, n\}$. For generation of random points with Halton distribution in higher dimension, different values of p are used for each dimension.

A. RBF Approximation with Stationary Points

It is known that stationary points are such points that hold equation

$$\frac{\delta f}{\delta x} = 0 \wedge \frac{\delta f}{\delta y} = 0 \quad (6)$$

This condition is not enough to determine whether the point is global/local extreme or just a saddle point. It is necessary to examine Hessian to determine point property. On the other hand it can be seen that saddle points are as important as points of extreme.

Evaluation of partial derivatives and comparing with zero is not optimal way to find stationary points. Better way is to compare given point with its surrounding i.e. masks for minimum, maximum and saddle point can be created.

B. RBF Approximation with Inflection Points

Points of inflection are such points where surface change from convex to concave or the other way round. For points of inflection in continuous space hold that Gauss curvature is equal to zero

$$\kappa_{gauss} = \frac{\frac{\delta^2 f}{\delta x^2} \frac{\delta^2 f}{\delta y^2} - \left(\frac{\delta^2 f}{\delta x \delta y} \right)^2}{\left(\left(\frac{\delta f}{\delta x} \right)^2 + \left(\frac{\delta f}{\delta y} \right)^2 + 1 \right)^2} \quad (7)$$

It can be seen, from equations above, that Gauss curvature is zero only when numerator is equal zero, i.e. Hessian matrix determinant is equal to zero.

It is not necessary to compute exact curvature value to find point of inflection. We need to find just points where curvature sign change from negative to positive or vice versa. It can be seen that sign of Gauss curvature only depends on numerator sign because denominator is always positive.

C. RBF Approximation with Stationary Points of Curvature

From (7) we can compute curvature of given surface in every data point. Then it is possible to find stationary points in curvature similar to process described in the section III-A

D. RBF Approximation with Edge Detection

In the simplified scenario, where the points are sampled in the grid pattern, we can look at the data as image i.e. value h on position $[x, y]$ can be considered to be brightness intensity I in pixel (i, j) . In case of scatter data there is need to use some special data structure to obtain points adjacency information e.g. kd-tree or adjust used algorithms.

We proposed to detect edges in image i.e. transitions between low and high values. Another suggested approach is to compute data gradient magnitude in each data point and then run edge detector over such field of gradient magnitudes. This approach will find transitions between low and high gradient magnitude areas. To detect edges we can use existing detectors from image processing e.g. Canny, Sobel, Prewitt etc.

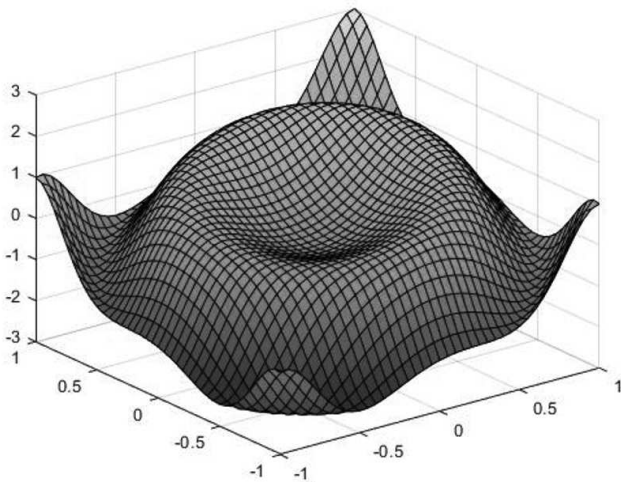


Fig. 1: Test function

IV. EXPERIMENTAL RESULTS

Proposed methods were tested on several test functions which were designed to represent special data set behaviour. Sampling step is 0.01 and functions are normalized to interval $(x, y, z) \in \langle -1, 1 \rangle \times \langle -1, 1 \rangle \times \langle 0, 1 \rangle$ for comparison purposes. For all functions Gauss radial basis function $\phi(r) = e^{-\epsilon r^2}$ was used.

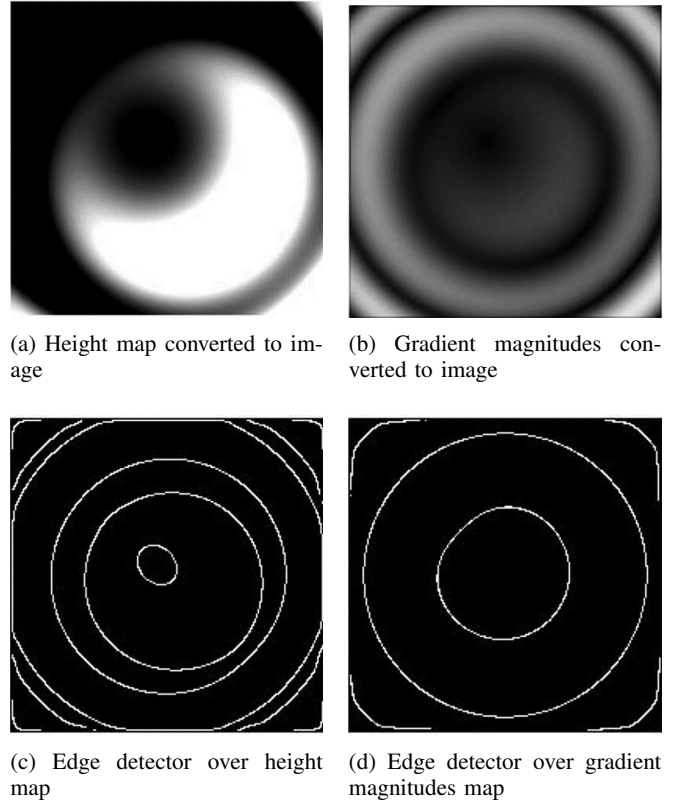


Fig. 2: The edge detection from height map (a), (c). The edge detection from gradient magnitudes map (b), (d).

$$\begin{aligned}
 f_1(x, y) &= \frac{2}{11} \left(\sin(4x^2 + 4y^2) - x + y - \frac{5}{2} \right) \\
 f_2(x, y) &= \frac{3}{4} e^{-\frac{1}{4}((9x-2)^2 + (9y-2)^2)} \\
 &\quad + \frac{3}{4} e^{-\frac{1}{49}((9x+1)^2 - \frac{1}{10}(9y+1)^2)} \\
 &\quad + \frac{1}{2} e^{-\frac{1}{4}((9x-7)^2 + (9y-3)^2)} - \frac{1}{5} e^{-(9x-4)^2 - (9y-7)^2} \\
 f_3(x, y) &= \frac{1}{9} \tanh(9y - 9x) + 1
 \end{aligned} \tag{8}$$

For comparison was used square mean error per point as can be seen in Fig. 6, Fig. 7 and Fig. 8. In the 1st test function (8) random distribution of placement with Halton sequence provide good results in comparison with other methods, see Fig. 3. This is caused by function shape which fill whole space.

In the 2nd test function (8) can be seen improvement when proposed methods are used because of its special behaviour only in some areas of its domain, see Fig. 4.

The last test function (8) is design to test RBF approximation in general and even with our improvements lot of methods fails, see Fig. 5. What is even more it was found that with proper placement it has no effect on precision to add more points from Halton sequence.

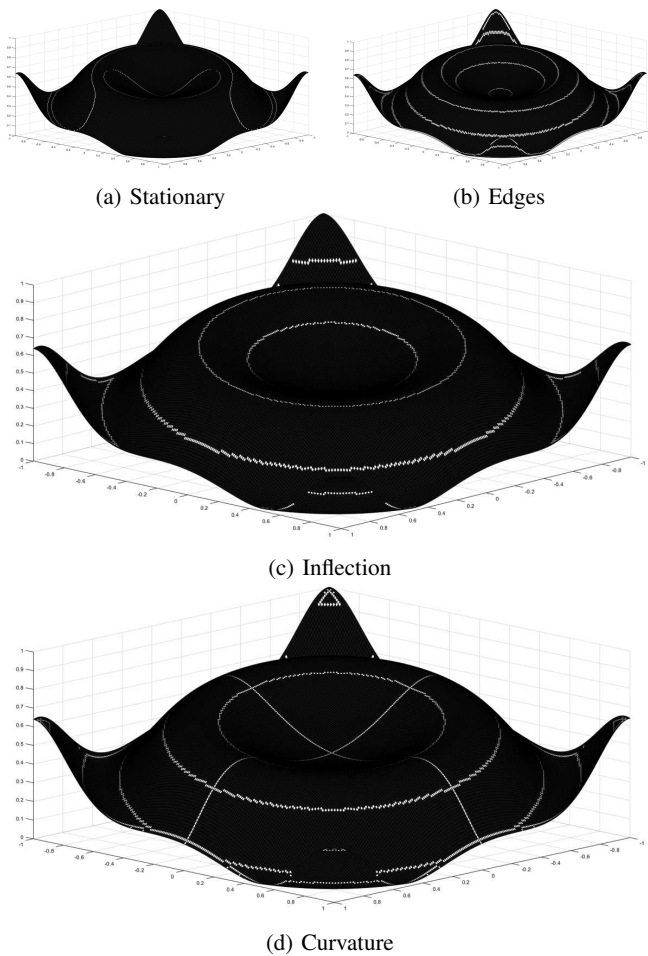


Fig. 3: Methods applied on 1st test function

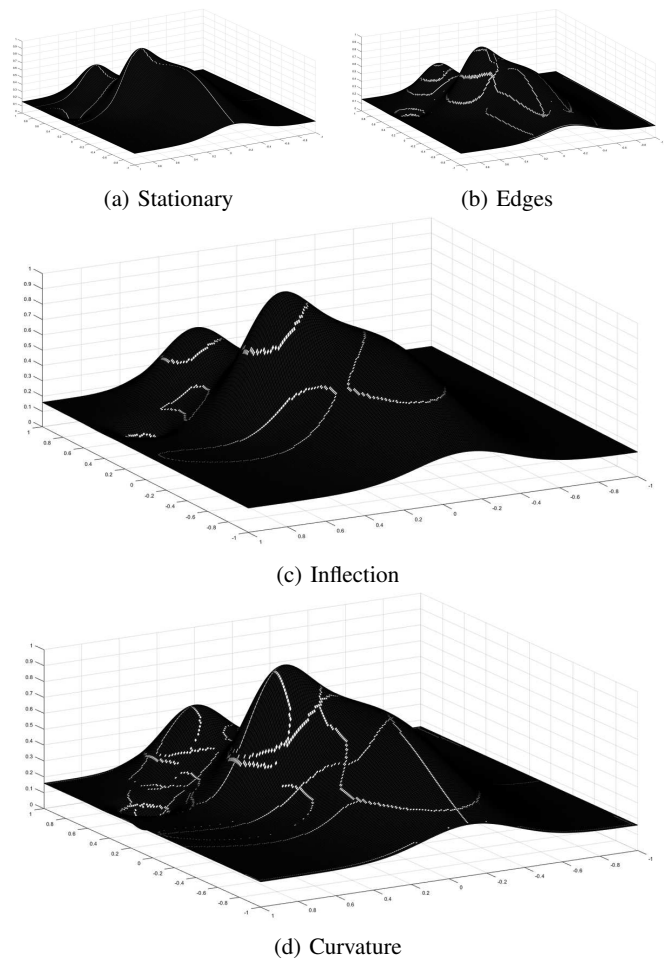


Fig. 4: Methods applied on 2nd test function

V. CONCLUSION

The proposed methods were tested on several standard testing functions, however, only some representative functions are mentioned in this contribution. The above presented methods proved very good results in precision of approximation, even though some special types of functions, e.g. fast changes, are problematic for all approaches. The experiments also proved validity of the proposed methods for the Radial basis function approximation of scattered data, with regard to a low approximation error with high points reduction leading to a high compression ratio.

ACKNOWLEDGMENT

The authors would like to thank their colleagues at the University of West Bohemia, Plzen, especially to colleague Zuzana Majdisova, for their discussions and suggestions. This research was supported by the projects: Czech Science Foundation (GACR) No. GA17-05534S and partially by SGS 2019-016.

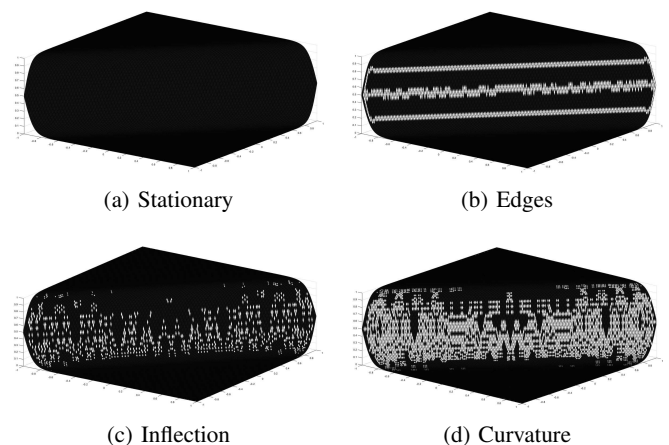
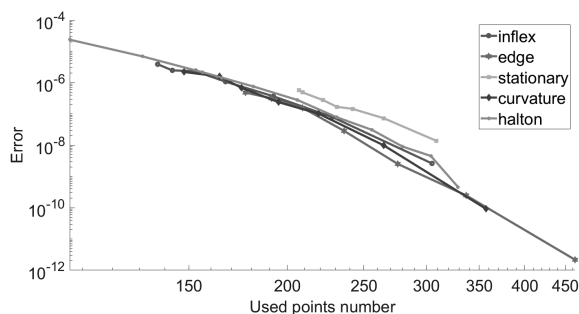
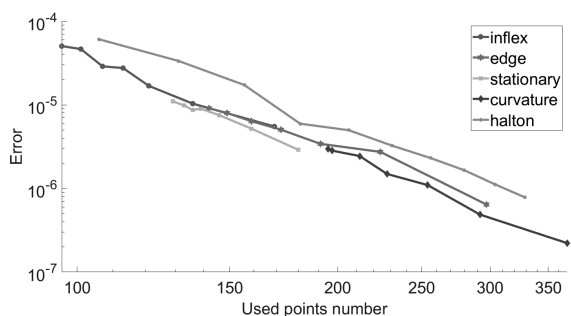
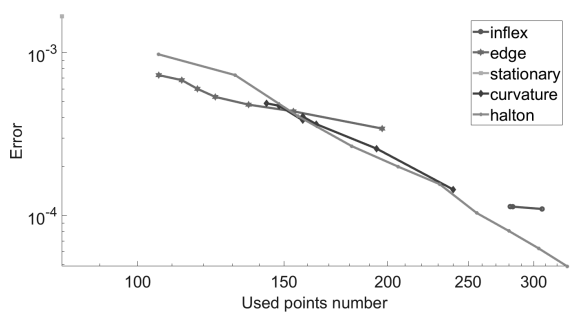


Fig. 5: Methods applied on 3rd test function

Fig. 6: Mean square error on 1st test function (8)Fig. 7: Mean square error on 2nd test function (8)Fig. 8: Mean square error on 3rd test function (8)

REFERENCES

- [1] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and mathematics III*. Springer, 2003, pp. 35–57.
- [2] M. Garland and P. S. Heckbert, "Fast polygonal approximation of terrains and height fields," 1995.
- [3] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," in *ACM Transactions on Graphics (ToG)*, vol. 23, no. 3. ACM, 2004, pp. 905–914.
- [4] C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3d scans," in *Computer graphics and interactive techniques*. ACM, 1995, pp. 109–118.
- [5] H. Li and S. S. Mulay, *Meshless methods and their numerical properties*. CRC press, 2013.
- [6] M. A. Oliver and R. Webster, "Kriging: a method of interpolation for geographical information systems," *International Journal of Geographical Information System*, vol. 4, no. 3, pp. 313–332, 1990.
- [7] I. Kaymaz, "Application of kriging method to structural reliability problems," *Structural Safety*, vol. 27, no. 2, pp. 133–151, 2005.
- [8] S. Sakata, F. Ashida, and M. Zako, "An efficient algorithm for kriging approximation and optimization with large-scale sampling data," *Computer methods in applied mechanics and engineering*, vol. 193, no. 3-5, pp. 385–404, 2004.
- [9] V. R. Joseph, Y. Hung, and A. Sudjianto, "Blind kriging: A new method for developing metamodels," *Journal of mechanical design*, vol. 130, no. 3, p. 031102, 2008.
- [10] A. Amiri-Simkooei and S. Jazaeri, "Weighted total least squares formulated by standard least squares theory," *Journal of geodetic science*, vol. 2, no. 2, pp. 113–124, 2012.
- [11] T. Asparouhov and B. Muthén, "Weighted least squares estimation with missing data," *Mplus Technical Appendix*, vol. 2010, pp. 1–10, 2010.
- [12] M. Smolik, V. Skala, and O. Nedved, "A comparative study of lowess and rbf approximations for visualization," in *ICCSA*. Springer, 2016, pp. 405–419.
- [13] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*. World Scientific, 2007, vol. 6.
- [14] Z. Majdisova and V. Skala, "Radial basis function approximations: comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728–743, 2017.
- [15] —, "Big geo data surface approximation using radial basis functions: A comparative study," *Computers & Geosciences*, vol. 109, pp. 51–58, 2017.
- [16] M. Smolik and V. Skala, "Large scattered data interpolation with radial basis functions and space subdivision," *Integrated Computer-Aided Engineering*, vol. 25, no. 1, pp. 49–62, 2018.
- [17] —, "Efficient simple large scattered 3d vector fields radial basis functions approximation using space subdivision," in *ICCSA*. Springer, 2019, pp. 337–350.
- [18] M. Cervenka, M. Smolik, and V. Skala, "A new strategy for scattered data approximation using radial basis functions respecting points of inflection," in *ICCSA*. Springer, 2019, pp. 322–336.
- [19] V. Skala and S. Karim, "Finding points of importance for radial basis function approximation of large scattered data," in *Symposium Kebangsaan Sains Metemakik ke 27*, 2019.
- [20] J. Wang and G. Liu, "On the optimal shape parameters of radial basis functions used for 2-d meshless methods," *Computer methods in applied mechanics and engineering*, vol. 191, no. 23-24, pp. 2611–2630, 2002.
- [21] J. González, I. Rojas, J. Ortega, H. Pomares, F. J. Fernandez, and A. F. Díaz, "Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1478–1495, 2003.
- [22] S. A. Sarra and D. Sturgill, "A random variable shape parameter strategy for radial basis function approximation methods," *Engineering Analysis with Boundary Elements*, vol. 33, no. 11, pp. 1239–1245, 2009.
- [23] Z. Majdisova, V. Skala, and M. Smolik, "Near optimal placement of reference points and choice of an appropriate variable shape parameter for RBF approximation," *Integrated Computer-Aided Engineering*, p. 16, 2019.
- [24] V. Skala, S. Karim, and M. Zabran, "Radial basis function approximation optimal shape parameters estimation: Preliminary experimental results," in *Symposium Kebangsaan Sains Metemakik ke 27*, 2019.
- [25] K. Uhlir and V. Skala, "Reconstruction of damaged images using radial basis functions," in *European Signal Processing Conference*. IEEE, 2005, pp. 1–4.
- [26] J. Zapletal, P. Vaněček, and V. Skala, "Rbf-based image restoration utilising auxiliary points," in *Computer Graphics International Conference*. ACM, 2009, pp. 39–43.
- [27] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.
- [28] R. Pan and V. Skala, "Continuous global optimization in surface reconstruction from an oriented point cloud," *Computer-Aided Design*, vol. 43, no. 8, pp. 896–901, 2011.
- [29] —, "A two-level approach to implicit surface modeling with compactly supported radial basis functions," *Engineering with Computers*, vol. 27, no. 3, pp. 299–307, 2011.
- [30] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of geophysical research*, vol. 76, no. 8, pp. 1905–1915, 1971.
- [31] C. A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions," in *Approximation theory and spline functions*. Springer, 1984, pp. 143–145.
- [32] G. B. Wright, "Radial basis function interpolation: numerical and analytical developments," 2003.

Towards a formal description and automatic execution of production processes

Marko Vještica
University of Novi Sad
Faculty of Technical Sciences
 Novi Sad, Serbia
 marko.vjestica@uns.ac.rs

Vladimir Dimitrieski
University of Novi Sad
Faculty of Technical Sciences
 Novi Sad, Serbia
 dimitrieski@uns.ac.rs

Milan Pisarić
Industrial Automation
KEBA AG
 Linz, Austria
 pisa@keba.com

Slavica Kordić
University of Novi Sad
Faculty of Technical Sciences
 Novi Sad, Serbia
 slavica@uns.ac.rs

Sonja Ristić
University of Novi Sad
Faculty of Technical Sciences
 Novi Sad, Serbia
 sdristic@uns.ac.rs

Ivan Luković
University of Novi Sad
Faculty of Technical Sciences
 Novi Sad, Serbia
 ivan@uns.ac.rs

Abstract—Industry 4.0 enforces digital revolution in the manufacturing industry by introducing smart devices and knowledge bases to form intelligent manufacturing information systems. One of the goals of the digital revolution is to support an automatic production from product specification to allow faster reaction on the product changes and a lot size one production. In order to make this possible, a formal language to describe production processes is needed, together with a code generator for its models and an engine to execute the code on smart devices. In this paper we propose a research on Industry 4.0 manufacturing using a Model-Driven Software Development (MDS) approach to model production processes, use the models to generate instructions to smart devices and human workers, and gather a feedback from them during the process execution.

Keywords—Industry 4.0, production processes, knowledge bases, formal languages, Model-Driven Software Development

I. INTRODUCTION

Fourth industrial revolution has been triggered by technological advances and increasing customer need for highly customized products. This manufacturing revolution is called Industry 4.0 in Germany and Smart Manufacturing in United States. Both countries, and many others like Japan, China and Korea, have established national programs and research projects in the domain of smart manufacturing in order to define and implement its concepts [1]. Although there is a lot of invested research effort, many research questions are left unanswered and challenges not yet addressed. One important research topic that is not fully addressed yet is the identification of concepts that lead to the creation of a formal language to describe production processes which in turn would allow for easier specification and production of highly customized products.

In traditional manufacturing systems, switching between different products or different product variants requires either stopping the production to reconfigure equipment or having different production lines for each product [2] which infers additional costs to the manufacturer. On the other hand, the goal of Industry 4.0 is to provide the automatic and flexible production thus enabling mass customization of products [3]. Such production flexibility would allow different products or product variants to be produced simultaneously without the need for stopping the production to do a reconfiguration [2]. In order to enable such flexibility, all the reconfiguration of the production line has to be done “on the fly” by the automated mechanisms of the smart factory and based on the

production process specification. Thus, production process specifications need to be formally described and machine readable. Production processes need to be specified in the form of production process models by using a domain-specific modeling language that includes concepts for representing materials, products, services, devices, human workers, communication between them and all process steps and tasks needed to create a product. Because of an increase of production processes complexity in the Industry 4.0, production process models will help process designers to think about production processes at a higher abstraction level and be more focused on modeling production process steps in order to minimize errors during processes specification. On the other hand, mechanisms for automating the production coupled with the mobile robots and workbenches will enable automatic reconfiguration based on the instructions stemming from the process models.

To create production process models, a digital twin of a physical process is needed that will enable a process monitoring, real-time decision making and control [4]. A digital twin represents a virtual model of a physical object which can simulate the object behavior [5]. In that way it is possible to simulate production steps and to predict their impact on the product [6]. These simulations are highly utilized in Industry 4.0 to simulate products, robots and humans in order to reduce failures and optimize resource (e.g. energy) consumption [7]. Without simulations, a production failure would happen more often due to the production processes complexity. Also, resource consumption could be higher than it should be because one product can be produced in different ways and simulations help to find out an optimal production process for a specific product. Having a formal and machine-readable production process model will allow for more precise and domain-specific simulations and enable easier creation and execution of simulations [8].

Besides the production simulations and manufacturing flexibility, having a formal production process models would provide means for better integration of humans in production processes in a way that is prescribed by the Lean production principles [9]. A visualization of production process models will be necessary for humans to either supervise the production or participate in the production [10]. The visualized production process models can be used by: i) human supervisors for the process monitoring, and ii) human workers as a manual to execute production tasks. The

visualization of production process models can provide human workers with a better understanding of processes, which will minimize errors during process execution and process monitoring.

In the paper we establish a proposal for a research on creation of a framework for the formal description and automatic execution of production processes. The framework is centered on a production process modeling language that has to be designed and implemented. The language can be categorized as a Domain Specific Modeling Language (DSML) aimed to specify production processes. The main goal of the proposed research is to define a methodological approach and a software solution in which the Model-Driven Software Development (MDS) principles and DSMLs will be used to formally specify production processes, to automatically generate instructions for smart machines or human workers and to gather a feedback from them during the process execution.

Apart from Introduction, this paper is organized as follows. In Section 2 theoretical foundations of proposed research are summarized. A context of a smart factory in which a production process modeling language will be used is described in Section 3. In Section 4 we present requirements for modeling production processes in Industry 4.0 and an MDS approach for automatic execution of code generated from the models. Conclusions and future work on implementation of Domain-Specific Language (DSL) related to production processes in Industry 4.0 are presented in Section 5.

II. BACKGROUND AND RELATED WORK

There are many possibilities and methodologies to choose for the specification and automatic execution of production processes. The proposed approach will follow MDS principles, centered on a DSML that is specific to a certain domain of application, in this case a domain of production processes modeling and execution. Accordingly, the theoretical foundations are grouped in three fields: basic concepts of smart factories, principles of MDS, and DSMLs.

To create a smart factory, advanced technologies like Cyber-Physical System (CPS), Internet of Things (IoT) and Wireless Sensor Network (WSN) are introduced in the manufacturing [11], [12]. CPS is the core element of the Industry 4.0 and production processes. It is used to denote physical objects with embedded software and computing power, which transforms products into smart ones and enables smart and decentralized production [13]. This transformation provides significant resource and cost advantages compared with traditional production systems [14]. CPSs are the connection between physical and virtual manufacturing worlds [15], and they are aimed to enhance process control by synchronizing these two worlds and by hiding the feedback from sensors and actuators state changes [16]. An exchange of the feedback and the communication between CPSs in manufacturing systems is provided by IoT and WSN. The smart products and machines have to be fully networked and integrated with smart factory in order to require minimal manual interventions [17].

The smart factory is based on advanced technologies, smart resources, smart materials and smart storage that are commonly used in Industry 4.0. In this context the smart resource is a service provider, the one that will process

operations, and the smart material is a service consumer, the one that will be processed [13]. In order to produce highly customized products with many different products involved in the production, a routing of smart resources should be reconfigured dynamically to enable easy switching between different smart products, which is very hard to achieve in traditional manufacturing. To enable dynamic routing, all these smart resources and smart products need to collaborate and communicate with each other. Modeling of smart resources and their collaboration and communication by means of a process modeling language can be of great help for process designers in Industry 4.0 environment. The production process modeling is an important industrial informatics researches topic [18], but it is still not sufficiently covered [19]. Production processes modeling languages have to be highly used in Industry 4.0 because production processes are digitally supported, and the goal of Industry 4.0 is to create an integration of production processes and cyber-physical factory [14]. Another difficulty is automatic execution of process descriptions that would be nearly impossible. Also, simulations and human integration would require a lot of manual work from process designers to provide them with necessary information. These are the reasons why production processes will be increasingly engineered with virtual representation that requires abstract thinking and modeling using a tool [17]. The modeling of production processes will be crucial in order to understand, control and optimize process operations [18]. Consequently, the production cost will be lowered due to the lowering of production failures and energy costs, accompanied with the lowering of production time.

A smart factory production system needs to have more smart resources of different types, like robots, machines or human workers. Automated Guided Vehicles (AGVs) are intelligent automatic vehicles used for transport and dynamic routing. AGVs are important resources in smart factory and intelligent manufacturing because they improve production by increasing automation and efficiency, reducing labor costs and making production fast and safe [6]. An example of rich transportation network of seven AGVs used for creation of different product variants could be seen at [2]. The transportation of smart materials and smart products can use most of the production time [20], so a management of transport vehicles is very important. In order to transport smart materials and smart products or execute different operations on them, AGVs can be equipped with Radio Frequency Identification (RFID) reader to read or write to the smart products' or smart materials' RFID tags and to communicate with other smart resources via IoT [12], [21]. RFID tags enable a wireless identification and localization of smart products and smart materials, and event generation to track the production process (e.g. the material is picked or placed) [22], [23].

AGVs or other robots can read products' or materials' state and they can detect if any disturbance occur due to automatic manufacturing process in order to manage the error (e.g. send a manual to human worker to manually fix the problem) [10]. A detection of any disturbance during the production requires an error handling, which is very important because errors can occur in any process step and they need to be carefully managed and modeled [2].

Besides machines and robots, human workers may be also considered as smart resources. They are integrated

within the smart factory and they work on tasks in which robots are not applicable or they work as problem solvers. Human workers interact with a smart factory using smart phones, smart watches, tablets or smart glasses to send or receive different kind of information [1], [10]. In comparison to human workers, autonomous robots can work on different tasks more precisely and in places that human workers are not allowed to work [7]. Also, in some countries, laws limit human workers to do some jobs (e.g. to lift heavy objects), and robots are in charge to do these jobs. Except the tasks where human workers and robots work separated, there are also tasks in which they work together. In the context of smart manufacturing, where the collaboration between human workers and robots is important, cobots are being used. Cobots are designed to work and collaborate with human workers and to ensure their safety during the production [1].

Human workers: i) supervise robots work, ii) collaborate with them during the production, and iii) work on the tasks that robots are not capable of doing. Collaboration and exchange of information between a human and machine is not only one that exist, but also between humans and between machines. In the context of Industry 4.0, most of the communication will be between machines [21], [24]. Most of the described communications and collaborations between machines and human workers need to be modeled within the production processes in order to specify exchange of messages between them and their work in parallel, which will make a production of highly personalized products faster. Otherwise the production can only be sequential and without significant optimization.

Besides smart resources and smart materials, there are also smart storages that can be equipped with different sensors in order to track their state or inventory. This is important information to know because during a production smart resources need to transport smart materials to available smart storages that lack specific materials for the production. In order to connect and control smart storages, resources and materials, an automatic management of production processes requires all of them to work together and to be connected with a decision-making system [21]. A cluster of industrial computers named Orchestrator can: i) delegate instructions to different smart resources, ii) manage transportation of materials and products between storages, and iii) detect and configure new smart resources or reconfigure existing ones [2], [25]. Examples of a manufacturing orchestration can be found in the work of Loskyll [26] and Keddis [27].

In order to orchestrate the production, a knowledge of: i) smart resources and their set of skills, ii) smart materials and their properties (e.g. dimensions, shape, mass), iii) production processes steps, skills needed for their execution and different constraints that must be included during the process execution, and iv) a production logistic is needed. For example, smart material mass is important property because not all smart resources are capable of transporting and processing a heavy material [28], and Orchestrator needs to find all of the capable smart resources and match them with process steps that requires any work with the material. The presented knowledge is similar to the manufacturing capability conceptual model presented in [29]. Orchestrator communicates with a knowledge base that stores all that knowledge and a reasoning mechanism must be implemented. Similar to Orchestrator, a Smart Machine

Agent module is described in [11] with a real-time data perception, a pool of knowledge and rules, a reasoning mechanism and an executor.

During a production, smart resources, materials and storages generate data from their sensors and that feedback data is needed for data analysis in order to optimize production processes. Generated feedback data requires a lot of storage memory, so the data is sent to a big data storage [12]. Every RFID sensor will send data like storages' status, smart resources' or smart materials' states and process steps' completion status during the production. An Enterprise Resource Planning (ERP) system gather the feedback data from RFID sensors through smart resource that can read the data and store it in a feedback data storage. This will allow data analysis and process monitoring to increase production quality, optimize energy consumption, reduce production costs and detect problems to maintenance them [6], [21].

An asynchronous communication between smart resources and an ERP system and between smart resources and Orchestrator is needed to gather a feedback and to delegate instructions. A communication protocol and platform-independent interfaces must be integrated with intelligent information system of the smart factory. Asynchronous communication could be established with REST APIs [16] and communication with different smart resources through sensors, actuators and mobile devices could be established with industrial protocol OPC UA [10], [16]. REST APIs and OPC UA protocol will be used to send instructions to smart resources generated from production process models, which will enable process execution without manual interventions.

Model-driven (MD) paradigm assumes orientation on models at all stages of system development. A complex system consists of several interrelated models organized through different levels of abstraction and platform specificity. MD as a prefix is an umbrella term to indicate, beyond others: MD Engineering (MDE), MD Software Engineering (MDSE), MD Development (MDD), MDSD and MD Architecture (MDA) [30]. Main goals of MDSD include the increase of the development speed through automation and single point of system definition; increase in software quality through formalization; increase in component reuse and improved manageability of complexity through abstraction; greater domain expert inclusion in the development process; and better communication between different stakeholders in the software development process [31]. MDSD is usually centered on the DSMLs that can be seen as a specialization of a wider notion of DSLs [32], [33]. The advantage of DSMLs in comparison to General Purpose Modeling Languages (GPML) is the closeness to the domain under observation and appropriateness of modeling concepts that are used for the given modeling task. By using such a language, a domain expert or a user familiar with the domain is able to specify the solution faster, with fewer errors, using familiar concepts than it is the case with GPMLs. An application of MDSD and DSMLs in industry systems' integration is given in [31].

III. A CONTEXT OF A SMART FACTORY

An example of assembly smart factory is presented in Fig. 1. Smart materials are stored in a global storage. If some of materials are needed for the production, an AGV will

bring all needed materials to the smart shelf. During the production, another AGV will bring materials from smart shelf to smart material area. A human worker and a cobot will work together or work in parallel to create a product using materials from smart material area. The product is created at a smart table. A finished product will be transported to a finished product area using another AGV. All these smart storages, smart materials and smart resources are tagged with a RFID so they can generate different information and smart resources are equipped with a RFID reader. The human worker has a mobile device to send or receive different information. This smart factory is managed by Orchestrator that communicates with a knowledge base to gather all the knowledge about the production and smart factory. A feedback from RFID tags is gathered by ERP and stored in data storage for data analysis and optimization. The data storage will have various data that is gathered from different sources and a large volume of data must be stored [5]. An example of reconfigurable assembly system that uses two industrial robots and RFID tags to generate events can be found at [22].

In the context of presented smart factory, we identified that a DSML aimed at production process modeling is needed to fully describe all process steps, smart resources, smart materials, smart products, transportation and collaboration in order to generate instructions from the process models to the smart resources. This will make a bridge between process specification and process execution. The language will also provide easier production process specification by process designers. As a result, the process specification will have fewer faults and will be more precise, and the process will be more optimized.

IV. A PRODUCTION PROCESS MDSO APPROACH

Within the context of the smart factory described in the previous section, we propose creation of a DSML aimed at production process modeling and MDSO approach to generate and automatically execute code on smart resources. This approach will enable easier production of customized products with less faults and costs. In this section we will describe basic concepts of proposed approach.

A production process model needs to be presented as a workflow that is composed of different activities (i.e. tasks) that need to be executed in a defined order [18]. An order of activities can be specified: i) as a sequence – a strict order in which activities must be executed (e.g. drilling must be done before sanding), ii) as a parallelism – two or more activities

that can be executed in parallel (e.g. two holes can be drilled in parallel), iii) as a flow control – depending on the specified parameter values, different activities are executed and iv) as an iteration. There are also activities that are unordered – they cannot be executed in parallel, but the order of execution is not important. An activity, a smart resource and a smart material together constitute a production process step. Every process step needs information about a smart resource that needs to execute an activity on a smart material. Process steps activities could be operations, transportations, configurations or inspections and they can contain different parameters about activity completion criteria and acceptance criteria. Completion and acceptance criteria are related to quality assurance, which is important part of production processes. Many tests could be performed by human workers or robots in order to check if quality of a product is acceptable and this should also be modeled with a production process language. Also, different kind of faults could occur during a production. There is a need to manage these situations with an error handler and they have to be described by a production process modeling language.

Besides presented workflow, there are material, message and energy flows that need to be modeled. Every time a smart resource has to execute an operation on a smart material, information about a material location needs to be known. In every production process step, there needs to be information whether a material has to be acquired from storage or a material is a result from a previous production process step. If a material has to be acquired from a storage, it will require transportation activities. Otherwise, a material is already present, and operation can be executed. In order to automatically execute code generated from models, all activities must be generated precisely, including every transport needed. A message flow presents a communication and collaboration between smart resources. Both communication and collaboration between human workers, a human worker and machine, and between machines need to be specified with a production process modeling language. An energy flow should also be considered while modeling production processes. For example, some robots must be plugged in before performing operations; some of them have a battery which status needs to be tracked in order to charge it in time or other sources of energy could be used. Except the presented four types of flow, a production process modeling language should also support specification of process variations. Besides product variations in which similar products are different in only few details, there are

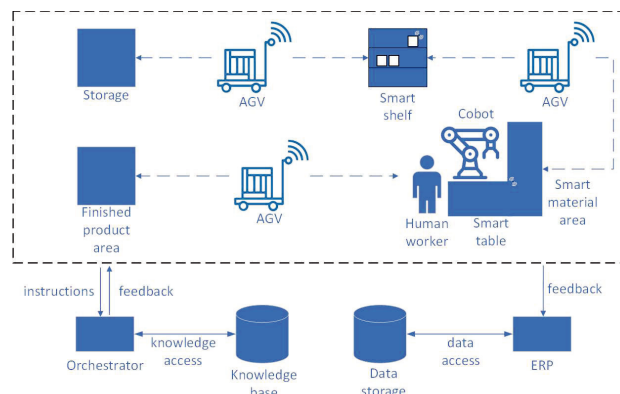


Fig. 1. An example of assembly smart manufacturing

also process variations in which the same product could be created using different process steps.

All presented production process modeling concepts are needed for an abstract syntax of a DSML in order to prepare models for code generation and automatic execution. Also, a concrete DSML syntax is needed. There are two types of concrete syntaxes: a textual and a graphical syntax, but there is no general answer which one is better [34]. A production process modeling language will be used by process designers and its models will be used by human supervisors and workers in a production. A graphical syntax is better option in this case, because it: i) enables visualized monitoring for human supervisors, ii) visualizes detected errors during a production, iii) enables graphical simulations of production process models, iv) creates manuals for human workers, and v) makes a modeling easier for production process designers. A production process monitoring has to be supported by a language to help human supervisors to track and control a production. Production errors also need to be visualized in order to easily detect in which production process step errors occurred. Visualization of production process simulations will help in finding out which production process steps have some defects and need to be optimized or production costs could be lower performing different changes on processes. A visualized production process models could also be used as a manual for human workers during a production. Human workers will receive a production process model on their mobile devices with a manual how to perform required activities. A graphical syntax will help process designers to make a modeling of production processes easier, but a language should also contain mechanism to deal with a production process complexity. Sub-processes will help to deal with large production process models to hide all details, but problems are also large amount of transportation activities and matching smart resources with every process step to fully support a production process for automatic execution. To help process designers not to think about transportation activities and smart resources, we propose a usage of Orchestrator and two abstraction levels of a production process modeling language. Process designers will use a language at a higher level of abstraction to specify production process steps that include an activity and material on which the activity will be performed. After the process steps are specified, Orchestrator will communicate with a knowledge base to identify which transportation steps are required and add them to a production process model. It will also identify which smart resources are able to execute specified activities related to every production process step and add these resources to the steps. After Orchestrator finishes a production process model at lower level of

abstraction, a code generator could be used to generate instructions to smart resources and a production could be started.

Using the described concepts of a production process modeling language and its graphical syntax, process designers can create different models that are ready for execution. The proposed MDS solution, in which a language is used, is presented in Fig. 2. Process designers can use a production process modeling language at a higher level of abstraction. These models can also be obtained from product descriptions (e.g. CAD models) using a process extractor. This process extractor uses knowledge extraction to conclude production process steps from a product description to generate a production process model. An example of process extraction from a digital product description to perform virtual assembly of the product could be found at [35]. Production process models at a higher level of abstraction will be used by Orchestrator that communicates with a knowledge base in order to add: i) transportation steps in a model, ii) smart resources to any production process step and iii) energy flow elements, which represents energy consumption of the smart resources. A knowledge base is used to match required skills for an activity specified in any production process step with skills of smart resources that are able to execute specified activity. Orchestrator generates models at a lower level of abstraction that are ready to be used by a code generator. Generated code needs to be human readable if instructions are sent to mobile devices of human workers or machine readable if instruction are sent to robots. During a production, feedback data is sent back to Orchestrator in order to monitor the production and detect if an error occurs to reconfigure production. Feedback data is also gathered by ERP and stored in data storage in order to analyze data and find out about possible optimization, energy or time leaking. Besides generating instructions for smart resources, the code generator also needs to be connected with a simulation. This way a generated code could be tested in a simulation before a production, so a production process model could be optimized or fixed if needed.

V. CONCLUSIONS

In previous sections we have identified and described requirements for modeling production processes in Industry 4.0 and preparing models for automatic production using an MDS approach. In order to prepare process models for automatic production, a DSML has to be specified. Based on our claims from the previous section, the DSML has to support: i) modeling of all production process concepts described in the previous section, ii) graphical syntax, iii)

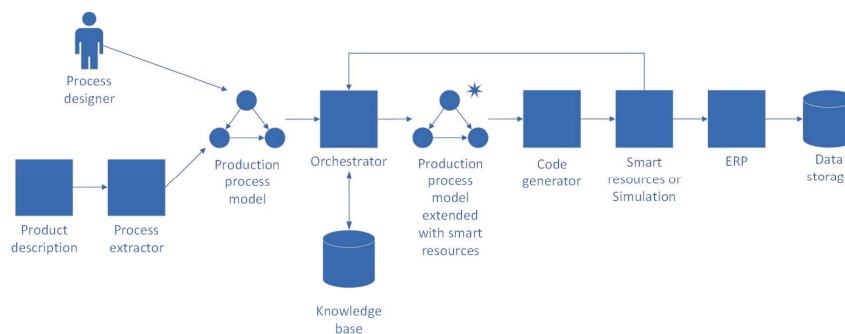


Fig. 2. A production process MDS approach

generation of human readable or machine readable instructions, iv) generating manuals for human workers, v) different levels of abstraction, vi) production process monitoring, vii) error handling, viii) dynamic changes of a model during a production, and ix) production simulations.

According to preliminary literature review, there are no languages that have all these characteristics and can create production process models that are ready for production. We propose the creation of a DSML that will support all production process requirements. The DSML will contribute in: i) better and faster production process implementation, ii) faster production process changes, iii) better integration of human workers, human supervisors and production processes, iv) lesser production faults and v) lower production time and costs. A modeling tool will be implemented to support process designer and an appropriate code generator will also be implemented to generate instructions and make automatic production possible. Created production process models can be used in different simulation scenarios to test or predict the behavior of production processes. This proposed approach will be tested using a simulation and in a real environment with smart resources and smart materials.

ACKNOWLEDGMENT

The research in this paper is supported by KEBA AG Linz and the Ministry of Education, Science and Technological Development of the Republic of Serbia, grant No. III-44010.

REFERENCES

- [1] K.-D. Thoben, S. Wiesner, and T. Wuest, "Industrie 4.0 and Smart Manufacturing – A Review of Research Issues and Application Examples," *Int. J. Autom. Technol.*, vol. 11, no. 1, pp. 4–16, Jan. 2017, doi: 10.20965/ijat.2017.p0004
- [2] K. Dorofeev, S. Profanter, J. Cabral, P. Ferreira, and A. Zoitl, "Agile Operational Behavior for the Control-Level Devices in Plug&Produce Production Environments," *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, 2019, pp. 49–56.
- [3] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, Jun. 2017.
- [4] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no. 5, pp. 616–630, Oct. 2017.
- [5] T. P. Raptis, A. Passarella, and M. Conti, "Data Management in Industry 4.0: State of the Art and Open Challenges," in *IEEE Access*, vol. 7, pp. 97052–97093, 2019, doi: 10.1109/ACCESS.2019.2929296
- [6] J. Wan, H. Cai, and K. Zhou, "Industrie 4.0: Enabling Technologies," *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, Harbin, 2015, pp. 135–140.
- [7] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0 – A Glimpse," *Procedia Manuf.*, vol. 20, pp. 233–238, 2018.
- [8] B. Rodič, "Industry 4.0 and the New Simulation Modelling Paradigm," *Organizacija*, vol. 50, no. 3, pp. 193–207, Aug. 2017.
- [9] D. Kolberg and D. Zühlke, "Lean Automation enabled by Industry 4.0 Technologies," *IFAC-Pap.*, vol. 48, no. 3, pp. 1870–1875, 2015.
- [10] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *12th IEEE International Conference on Industrial Informatics (INDIN)*, Porto Alegre RS, Brazil, 2014, pp. 289–294.
- [11] Y. Zhang, C. Qian, J. Lv, and Y. Liu, "Agent and Cyber-Physical System Based Self-Organizing and Self-Adaptive Intelligent Shopfloor," *IEEE Trans. Ind. Inform.*, vol. 13, no. 2, pp. 737–747, Apr. 2017, doi: 10.1109/TII.2016.2618892
- [12] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing Smart Factory of Industrie 4.0: An Outlook," *Int. J. Distrib. Sens. Netw.*, vol. 12, no. 1, pp. 1–10, Jan. 2016, doi: 10.1155/2016/3159805
- [13] F. Almada-Lobo, "The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES)," *J. Innov. Manag.*, vol. 3, no. 4, pp. 16–21, Jan. 2016, doi: 10.24840/2183-0606_003.004_0003
- [14] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2941–2962, Apr. 2018, https://doi.org/10.1080/00207543.2018.1444806
- [15] R. Neugebauer, S. Hippmann, M. Leis, and M. Landherr, "Industrie 4.0 - From the Perspective of Applied Research," *Procedia CIRP*, vol. 57, pp. 2–7, 2016, https://doi.org/10.1016/j.procir.2016.11.002
- [16] B. B. Sanchez, R. Alcarria, D. Sanchez-de-Rivera, and A. Sanchez-Picot, "Enhancing Process Control in Industry 4.0 Scenarios using Cyber-Physical Systems," *JoWUA*, vol. 7, pp. 41–64, 2016.
- [17] S. Erol, A. Jäger, P. Hold, K. Ott, and W. Sihn, "Tangible Industry 4.0: A Scenario-Based Approach to Learning for the Future of Production," *Procedia CIRP*, vol. 54, pp. 13–18, 2016.
- [18] L. D. Xu, "Enterprise Systems: State-of-the-Art and Future Trends," *IEEE Trans. Ind. Inform.*, vol. 7, no. 4, pp. 630–640, Nov. 2011.
- [19] R. Petrasch and R. Hentschke, "Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method," *13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, 2016, pp. 1–5, doi: 10.1109/JCSSE.2016.7748885
- [20] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 1–4, pp. 147–164, Apr. 2016.
- [21] V. Roblek, M. Meško, and A. Krapež, "A Complex View of Industry 4.0," *SAGE Open*, vol. 6, no. 2, pp. 1–11, Apr. 2016.
- [22] S. Makris, G. Michalos, and G. Chryssolouris, "RFID driven robotic assembly for random mix manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 3, pp. 359–365, Jun. 2012.
- [23] T. Stock and G. Seliger, "Opportunities of Sustainable Manufacturing in Industry 4.0," *Procedia CIRP*, vol. 40, pp. 536–541, 2016.
- [24] T. Wagner, C. Herrmann, and S. Thiede, "Industry 4.0 Impacts on Lean Production Systems," *Procedia CIRP*, vol. 63, pp. 125–131, 2017, https://doi.org/10.1016/j.procir.2017.02.041
- [25] M. Pisarić, V. Dimitrieski, M. Babić, S. Veselinović, and F. Dušić, "Towards a Plug-and-Play Architecture in Industry 4.0," *17th International Scientific Conference on Industrial Systems (IS'17)*, Novi Sad, Serbia, 2017, pp. 136–141.
- [26] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber, and B. Pirvu, "Semantic service discovery and orchestration for manufacturing processes," in *ETFA2011*, Toulouse, France, 2011, pp. 1–8, doi: 10.1109/ETFA.2011.6058988
- [27] N. Keddiss, "Capability-Based System-Aware Planning and Scheduling of Workflows for Adaptable Manufacturing Systems," Ph.D. dissertation, Technical University of Munich, Germany, 2016, ISBN: 978-3-86844-838-2
- [28] M. Trstenjak and P. Cosic, "Process Planning in Industry 4.0 Environment," *Procedia Manuf.*, vol. 11, pp. 1744–1750, 2017.
- [29] Y. Luo, L. Zhang, F. Tao, L. Ren, Y. Liu, and Z. Zhang, "A modeling and description method of multidimensional information for manufacturing capability in cloud manufacturing system," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 5–8, pp. 961–975, Nov. 2013.
- [30] T. Stahl, M. Voelter, and K. Czarniecki, *Model-Driven Software Development: Technology, Engineering, Management*, 1st ed. Hoboken, NJ: Wiley, 2006, ISBN 978-0-470-02570-3
- [31] V. Dimitrieski, "Model-Driven Technical Space Integration Based on a Mapping Approach," Ph.D. dissertation, Faculty of Technical Sciences, University of Novi Sad, Serbia, 2017.
- [32] A. van Deursen, P. Klint, and J. Visser, "Domain-specific languages: an annotated bibliography," *ACM SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, Jun. 2000, doi: 10.1145/352029.352035
- [33] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, Dec. 2005, doi: 10.1145/1118890.1118892
- [34] I. Dejanovic, M. Tumbas, G. Milosavljevic, and B. Perisic, "Comparison of Textual and Visual Notations of DOMMLite Domain-Specific Language," *Local Proceedings of the Fourteenth East-European Conference on Advances in Databases and Information Systems*, pp. 131–136, 2010.
- [35] S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions," *Comput. Ind.*, vol. 97, pp. 34–46, May 2018.

Possibilities of human interaction in the virtual environment

Patrik Voštinár

*Department of Computer Science,
Faculty of Natural Sciences,
Matej Bel University,
Banská Bystrica, Slovakia
patrik.vostinar@umb.sk*

Dana Horváthová

*Department of Computer Science,
Faculty of Natural Sciences,
Matej Bel University,
Banská Bystrica, Slovakia
dana.horvathova@umb.sk*

Martin Mitter

*Department of Computer Science,
Faculty of Natural Sciences,
Matej Bel University,
Banská Bystrica, Slovakia
martin.mitter@studenti.umb.sk*

Martin Bako

*Department of Computer Science,
Faculty of Natural Sciences,
Matej Bel University,
Banská Bystrica, Slovakia
martin.bako@studenti.umb.sk*

Abstract—Virtual, augmented and mixed reality (VR, AR and MR) is now enjoying popularity not only in gaming, but as surveys indicate, also in the health care, engineering, live events, entertainment, real estate, retail, military, education, etc. The aim of this article is to show various possibilities of human interaction with emerging technologies in three different areas: playing games, health care and education. There is a lack of software development experts for VR, AR and MR everywhere. Therefore, our intention at the Department of Computer Science, Faculty of Natural Sciences, Matej Bel University in Banská Bystrica, Slovakia, is to focus on the education and enlightenment of these areas. In this paper, we present several fields, which use different human interaction modes.

Keywords—Human Interaction · Immersion · Interactivity · Virtual and mixed reality · Virtual environment · Games · Phobia treatment · Education

I. INTRODUCTION

VR offers a unique personal experience that alters one's perception of the environment around them. While VR headsets have many mechanic limitations, the immersion is strong enough that the elusive goal of presence is reached [1]. The terms “presence” and “immersion” are discussed by a wide field of researchers. “Presence” generally refers to the sensation of “being there”, or as a subjective phenomenon such as the sensation of being in a virtual environment. “Immersion” as the extent to which the senses are engaged by the mediated environment, or an objective description of aspects of the system such as field of view and display resolution [2].

In the context of virtual reality, the term immersion is used to describe the user's emotional reaction to the virtual world in terms of feeling as if they are actually a part of the virtual world [3].

After decades of experimentation with VR beginning in the late 1980s, hardware, software, and consumer mind-sets are finally ready for the immersive VR experiences its early visionaries dreamed [4].

We also had a dream at the Department of Computer Science in Banská Bystrica in the beginning to help psychotherapists in phobia treatment. Therefore, we have become more interested in VR. After six years of research and investigation, we have come to the knowledge that we will implement in all areas that we address in this article.

The first field we have focused on is the field of games, which has always been, is and probably will be a pioneer in crossing borders in the development of the VR. Therefore, we would like to show in our research how simple game can be useful from the perspective of interactivity. We decided for a simple sport topic that can demonstrate the interactivity, physics and credibility of the virtual environment. In our contribution, we present the application of basketball, where throwing the ball into the basket will test the possibilities of human interaction with the virtual environment. It trains also movement skills and space coordination at the same time.

The second field is the usage of the VR for health care. Experience from several researches [5], [6], [7], [12], [13], [14] shows the success and effectiveness of treatment of some phobias with VR. We have developed several applications that can be used to treat different phobias. Here we focus on the treatment of one of the most widespread specific phobias, such as arachnophobia - the disease fear of spiders. The application is designed to be controlled not only by the patient, but also by the therapist. In this way, the therapist has a full process under control.

The third field is education. VR has good impact on education system. It makes the education system more entertaining, and students interact with each other in 3D environment and learn new things very fast as compared to traditional education system. Student can learn by doing fun in VR environment [6]. Our goal in this field is to prepare our students to become future developers of the VR, AR and MR. In this article, we focus on the course of VR in LMS Moodle. Our students here have available lessons not only to develop a virtual environment, but they will also work with a panoramic camera, as well as with the creation of MR and AR.

Thanks to our research, projects and cooperation with companies we have the necessary software and hardware with which we started to develop applications, such as e.g. HTC Vive with a powerful PC, with a dedicated graphics card. A pair of HTC Vive controllers, whose movement is transmitted to the PC using a couple of Base Station sensors, ensure interactivity.

II. VIRTUAL REALITY IN GAMES

According to [7] a “game” is a system within which players traditionally engage in an artificial conflict, trying to solve a specific problem. A game is defined by rules and

measured by a quantifiable outcome. Usually the game has several key elements:

- A specific goal that people are willing to work for.
- Rules that stimulate creativity.
- A feedback system that lets individuals know how they are doing with respect to the goal.
- Voluntary acceptance of the goal, rules, and feedback systems.

VR and digital games (DG) are two areas, which share many similar characteristics. Both of them have to focus on human to succeed (feeling of presence for VR, entertainment for DG). VR and DG exploit the technological breakthroughs of several fields like image synthesis, electronics, etc. And finally, they both act sometimes in a VR world like a flight simulator or therapy environment. Other times they allow operating in a fantasy world which does not necessarily respect usual physical laws [8]. Though modern technological approaches to VR originated in gaming, its applicability extends beyond those platforms [9]. Many principles and techniques in conventional computer game design allow for creating immersive, interesting, and fun games [10]. Virtual reality gaming is where a person can experience being in a three-dimensional environment and interact with that environment during a game. This is an essential part of the game [11]. Since all players want to be immersed when playing the games, in particular when playing in virtual environments and the feeling of embedding can be strongly influenced by interactivity with the environment and its objects, we also focus in our article on ways to achieve delusion of the senses.

A. Application Basketball

Our first application we want to introduce is basketball. It is an interactive virtual environment that looks like a sports field. There is a basketball court on the playground with a grass (or concrete) floor. There are several basketball balls in this environment that randomly roll around the environment after loading. Object that enters the environment has the opportunity to grab the ball, throw it, or hit the basket. The ball collides with other balls and after an accidental collision with them, these balls move, allowing for functional and believable physics. If you hit the ball in the basket, the points are scored on the screen. There is a button on the desktop that resets the game to the default state when you press it.

B. Creation tools

We used the software Unreal Engine to create the virtual environment of the application and adapted the project to the mentioned HTC Vive hardware. We put the player's camera in the new level, which contained virtual reality and collision controllers. Then we placed a surface with a texture of 3D grass under the camera. In Cinema 4D, we modelled the basket and ball (textured from NBA 2002) and then imported them into the Unreal Engine. Here we set collisions so that the ball could fall through the hoop so that it did not remain stuck at the top of the basket, or to bounce if the hit was not accurate and so on. With the ball, it was necessary to adjust the dynamic physics with respect to gravity, the angle of incidence of the ball and, of course, to reflect the ball from the surface as naturally as possible. Fig. 1 shows software Unreal Engine with open project Basketball.

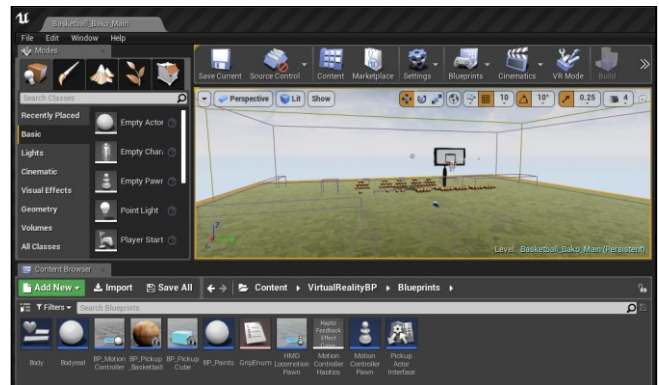


Fig. 1. Software Unreal Engine and Basketball project.

C. Interactivity

It was also necessary to solve the movement around the environment and the possibility of catching the ball. We solved the movement using the buttons of HTC Vive controllers, when the player presses the button, holds it and points where he wants to go. After it is released, the player will teleport to that location. The grip of the ball was solved using the trigger release button. When it was pressed, the player was able to bounce it, but as soon as he was close to it, he pulled the trigger, the player grabbed the ball and was able to throw it into the basket. The last step was to adjust the collisions of the ball with the inner space of the hoop to add the points that appear on the screen. We also added a button to the environment where the player reset the level when he threw the ball. Fig. 2 shows interactivity in Basketball application.



Fig. 2. Example of created VR game – Basketball.

III. PHOBIA TREATMENT BY HELP OF VIRTUAL REALITY

Clinical psychology describes phobias as an anxiety disorder, characterized by an intense irrational fear of specific objects or situations. This excessive amount of fear does not correspond to the potential amount of danger of stimulus. Despite the fact, people suffering from phobias experience

intensive psychic symptoms (anxiety, loss of control, fear) and vegetative symptoms (increased heartbeat, fainting, sweating, problems with breathing) etc. [12], [13], [14].

Generally, phobias could be divided into three main categories: agoraphobias, social phobias, and specific phobias [15].

A. Application Arachnophobia

In designing phobia treatment applications, we introduce the VR application for HTC Vive, which focuses on the treatment of one of the most widespread specific phobias, such as arachnophobia - the disease fear of spiders. The purpose of the application is to streamline the course of therapy and provide a tool to tailor the entire process according to specific requirements.

B. Creation tools

The application was created on the VR Ready PC, which was connected to HTC Vive. During its creation, several software products were used, but the core of the application runs on the game engine Unity 3D. We used Blender to model additional objects, and we edited textures in Adobe Photoshop. Graphics and buttons were created in Adobe Illustrator. The whole logic of the application is programmed in the C# scripting language using scripts that are assigned to individual objects in the scene. These are scripts for random movement of the spider, avoiding obstacles, touching and catching objects, killing the spider, controlling the GUI elements and so on.

C. Interactivity

When you open the application, the main menu appears in which we have several options: launching the main part of the application, description of the operation, settings, as well as information about the application and its author. The menu is also enriched with a model of animated spider, whose task is to clarify a smile on his face rather than induce phobic states.

The application is designed to be controlled not only by the patient but also by the therapist. In this way, the therapist is in control of the entire course of treatment. For the patient, interactivity is ensured by a pair of HTC Vive controllers that move to the PC using a couple of Base Station sensors. The therapist can use the keyboard and mouse to interfere with the application. We have programmed the application controls to be as simple and intuitive as possible for the user. The original controllers' model that come with the VIVE Input Utility (VIU) library have been replaced by realistic-looking hand models. These include animations of idle, touching, catching, and clenching. From a programmatic point of view, we have transformed these movements into a functional interaction with the surrounding objects. We used the trackpad button to touch the object, and we decided to use the grip button to catch and release, as it is the most natural button for the task. Killing the spiders and confirming the choice we mapped to trigger. The last function button is the menu button, which activates and deactivates the laser pointer.

D. Patients and therapists controlling

The user interface of the application is divided into two-dimensional and three-dimensional interface. Two-dimensional (Non-Diegetic UI) is only displayed on the PC monitor. It serves the therapist to control the elements of the environment at the discretion and response of the patient. The three-dimensional interface (Diegetic UI) has the ability to see

both, but only the patient can modify the environment objects through it. The advantage of such a divided user interface is that the patients can use the application in the comfort of their own home, which can significantly shorten the treatment process, not to mention the money saved while sitting with the therapist.

By clicking the start button on the main menu, the patient finds himself in a virtual environment to create a calm atmosphere and eliminate stress. Directly opposite the patient is a TV set that serves as an initial phase of therapy. On the TV screen, the patient displays images of spiders, from a small cartoon spider to a real photograph of a tarantula. Individual levels are used to eliminate unwanted or exaggerated patient reactions. By humorously beginning therapy, we encourage the patient not to be afraid to proceed to the next levels. The advantage is that if necessary, e.g. when we adapt an app to a different type of phobia, we can easily replace those images with another.

On the left side, the patient can see a whiteboard on which the user interface elements are located. There are buttons, checkboxes, dropdown menu and slider. These elements are used to add instances of spiders to the table, remove, enlarge, shrink, discolor, or, in the case of impulsive patient response, to remove all spiders. We implemented the spider instances into the stack data structure (FIFO). Each spider model is assigned a random animation using the Mecanim state machine and the corresponding script. By random generation, we have ensured that each spider behaves differently at a given moment, thereby achieving greater naturalness. Fig. 3 shows interactivity in Arachnophobia application.

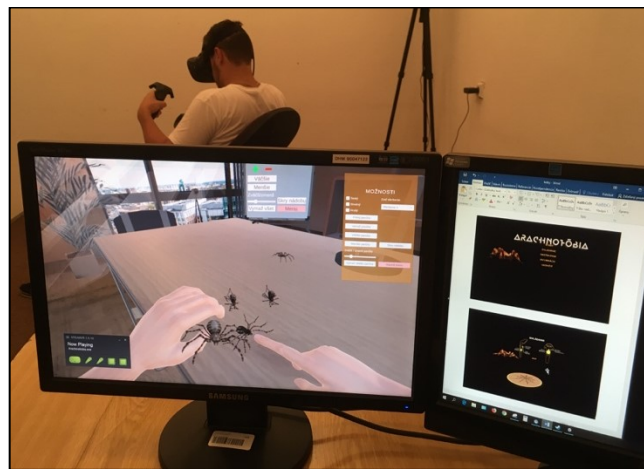


Fig. 3. Interactivity in application Arachnophobia.

The visual aspect of the application is complemented by sound effects that contribute to a greater immersion of the patient into the virtual environment, which in turn helps to increase the effectiveness of the treatment.

The application itself is also made more attractive by the elements of gamification, which move it to a higher level. We decided to incorporate the play elements to make the treatment process more interesting, interactive and engaging. Another reason is a more attractive form of treatment, encouraging active participation and engagement, allowing for more direct and immediate feedback. Information on how the patient was doing during therapy (the number of spiders touched, caught and killed) motivates him to progressively improve, thereby achieving his goal of getting rid of his unjustified fear.

In the future, we will be able to extend the application with other types of phobias. By simply exchanging animal models and their animations, we can create, for example, an application for the treatment of one of the zoophobias: musophobia, ophidiophobia, etc. There is also the possibility of targeting other VR platforms.

IV. VIRTUAL REALITY IN EDUCATION

In recent years, the media has presented VR and AR as technologies that reach home through various electronic devices such as helmets, goggles, smartphones, etc. This will involve the arrival of these technologies in educational environments by supporting different learning styles, and easing teaching and learning processes [16]. Since 2016, the KI FPV UMB in BB has been teaching the subject of VR, where students become acquainted with the area not only of VR, but also AR and MR. To support the subject, an electronic course in LMS Moodle is offered, in which students can find presentations, links to various websites and materials. Using video tutorials, they create assignments to build VR applications in Unity 3D, panorama video and photography, to combine models and panoramic video, and to create AR applications too. Fig. 4 shows small part of main menu of LMS Moodle course for VR.

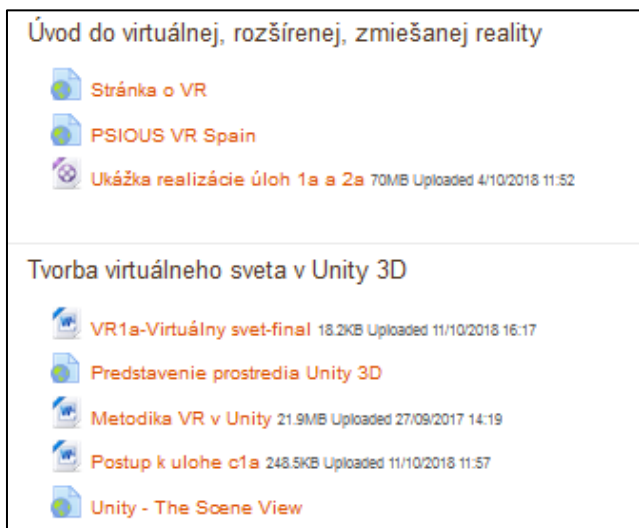


Fig. 4. LMS Moodle course for VR.

A. Combination of panoramic video and animated model

One of the interesting and challenging tasks that students do, is to create a mixed reality, where they create a combination of a real-world, captured by a panoramic camera with animated models. Such combination brings new possibilities of using virtual graphic not only in area of games and entertainment, but it can also help in area of health and well-being, when real looking environment could create a feeling of dreadful situation. Fig. 5 shows such combination of forest environment in which is placed an animated deer model.

B. Hardware and software tools

Panoramic videos can be created using special cameras with wide-angle lenses. They can take a full 360-degree panorama in both horizontal and vertical directions. In our case, we use Samsung Gear 360 for video recording that is simple, user-friendly and affordable. The camera captures two different spherical images into one file, which must then be combined

into a so-called „panoramic format“. We use Gear 360 Action Director software that comes together with the panoramic camera. We use Blender 3D software and its advanced motion tracking functions to create a mixed reality that combines panoramic video with an animated object.



Fig. 5. Combination of forest environment with an animated deer model.

C. The process of creating

The process of creating a MR in this software consists of several steps. First, markers had to be placed on the contrast points in the movie - the number of them depends on the video capture technique. After placing enough tags, we tracked them. An important step was to set the camera parameters. We had to ensure that the virtual camera in the program had the same parameters as the camera with which we recorded the video. Then the program calculated the movement of the traced tags. The calculation resulted in information on the size of the deviation. In the next step, we prepared the scene so that the 3D object could be placed in the video, which included dividing the scene into the foreground and background. We then imported a 3D model into the scene that included the armature and animations. We placed the loaded model in the desired position, rotated it correctly, and resized it to fit the video exactly. We provided lighting for the scene using two lighting methods, which combined to create the same lighting conditions as in reality. The first method is IBL (Image-Based Lighting), which uses image information from real photography to illuminate. As a second method, we used the virtual illumination found in the program. Finally, we rendered the scene into the resulting panoramic video enriched with an artificial 3D object.

D. Interactivity

An essential feature of the panoramic video is its interactivity - you decide for yourself what you are interested in at the moment (you choose the direction and angle of view). There are several ways to interact with the image. We can talk about the so-called „an immersive“ and „non-immersive manner“ of interaction, respectively presentation of the video. An immersive way is possible with one of the HMDs (HTC Vive, Samsung Gear VR, Google Cardboard, etc.) that responds to head movements or direction of vision and automatically adjusts image rotation. The non-immersive way is represented by software players (mobile or desktop) in which we interact with the video with a finger or mouse.

E. The possibilities of using

The possibilities of using MR (a combination of panoramic video and animated 3D object) are seen mainly in the film industry, in the creation of film tricks and visual

effects (shortly VFX). However, it is also irreplaceable in medicine in the treatment of various kinds of phobias by the exposure method. In this way, environments or situations can be created that would be difficult or impossible to capture in the real world. Thus, we can adapt the filmed environment and 3D objects exactly to the needs for treating phobia in a particular patient.

V. CONCLUSION

Since we are only at the beginning, we have not finished research for all these fields.

The first field is easy to test because of students in the school environment. Developed games have a positive response among students. Whether they are sports games or an escape rooms, we try to interact with the environment and its objects everywhere.

In the second field, we do not have feedback from real patients; the test phase is carried out only under laboratory conditions on simulated patients. The use of data helmets and gloves (or controllers) is unpleasant for some users (especially those who suffer from phobia), but research to date suggests that this feeling is better than touching the real object of their phobia. So adapting to a phobia object via VR is easier and not so frustrating.

We use the third field of VR, AR, MR in education mostly because we are an educational institution, which wants to keep up with modern emerging technologies and has the ambition to closely link the process of education with practice. It is the field of education that unites our efforts to create various applications for these areas, which we have the opportunity to test in real.

ACKNOWLEDGMENT

This contribution has been processed as part of the grant project Implementation of Blended Learning into Preparation of Future Mathematics Teachers, project no. 003UMB-4 / 2017th, project Interactive Applications for Teaching Mathematics at Primary Schools, project no. 003TTU-4/2018, KEGA no. 009KU-4/2017 Innovative methodology in the subject of computer science in secondary education. This contribution has been supported also by the Foundation of Tatra Banka (grant program E-Talent).

The authors would like to thank Nettech, Ltd., which provided hardware for experiments and therapist Mr. Ján Zásalkan for his valuable comments and suggestions to improve the quality of our research.

REFERENCES

- [1] G. Hodgkinson, "Lock up your stories – here comes Virtual Reality," in *Journal of Arts and Imaging Science*, vol. 3, no. 4, pp. 10-14, 2016.
- [2] G. Hodgkinson, "A New Medium for Animation – Stereo Virtual Reality," in *Journal of Arts and Imaging Science*, 2015.
- [3] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, A. Walton (2008). "Measuring and Defining the Experience of Immersion in Games", *International Journal of Human-Computer Studies*. Harvard: Elsevier.
- [4] Ch. Stein, "Virtual reality design: How upcoming head-mounted displays change design para-digms of virtual reality worlds," in *MediaTropes eJournal*, vol. VI, pp. 52–85, 2016.
- [5] A. 'Skip' Rizzo, R. Shilling (2017) "Clinical Virtual Reality tools to advance the prevention, assessment, and treatment of PTSD", *European Journal of Psychotraumatology*, 8:sup5, Available at <https://doi.org/10.1080/20008198.2017.1414560>
- [6] A. Bernardo, "Virtual Reality and Simulation in Neurosurgical Training," *World Neurosurgery*, vol. 106, pp. 1015-1029, 2017.
- [7] B. U. Stetina, A. Felhofer, O. D. Kothgassner, and M. Lehenbauer, "Games for Health: Have Fun with Virtual Reality!" in *Virtual Reality in Psychological, Medical and Pedagogical Applications*, Rijeka, Croatia: InTech, pp. 65-80, 2012.
- [8] P. Bouvier, F. Sorbier, P. Chaudeyrac, V. Biri, "Cross benefits between virtual reality and games," in *International Conference and Industry Symposium on Computer Games; Animation, Multimedia, IPTV, Edutainment and Security (CGAT'08)*, France, 10pp, 2008.
- [9] P. Rubin, "Oculus Rift," in *Wired*, vol. 22, no. 6 (2014), 78.
- [10] J. Schell, „The Art of Game Design, A Book of Lenses”, 2nd ed. Wellesley: A K Peters/CRC Press, 2014.
- [11] Virtual Reality Society, 2017, "What is virtual reality gaming?" Last access October 2019 Available at <http://www.vrs.org.uk/virtualreality-games/what-is-vr-gaming.html>.
- [12] D. Horváthová and V. Siládi, "Creating virtual environments for phobia treatment," in *Open Computer Science*. 2016; vol. 6, pp. 138–147.
- [13] A. Heretik, "Anxiety (neurotic) disorders," in *Clinical psychology, Nové Zámky, Psychoprof, spol. s.r.o.*, pp. 217–241, 2007.
- [14] S. Grenier, H. Forget, S. Bouchard, S. Isere, S. Belleville, O. Potvin, et al., "Using virtual reality to improve the efficacy of cognitive-behavioral therapy (CBT) in the treatment of late-life anxiety: Preliminary recommendations for future research," in *International Psychogeriatrics*, vol. 27, pp. 1217–1225, 2015.
- [15] "International classification of diseases", 2013. [Online]. Available: <http://www.nczisk.sk/Standardy-v-zdravotnictve/Pages/Medzinarodna-klasifikacia-chorob-MKCH-10.aspx> [Accessed: September 11, 2015]
- [16] J. Martín-Gutiérrez, C. E. Mora, B. Añorbe-Díaz, A. González-Marrero, "Virtual Technologies Trends in Education" in *EURASIA Journal of Mathematics Science and Technology Education*, ISSN 1305-8223 (online) 1305-8215 (print), 2017, 13(2):469-486

A proposal for service design based on user's action history using machine learning

Xinyue Wang*, Nobutada Fujii, Toshiya Kaihara, Daisuke Kokuryo.
Department of System Science, Graduate School of System Informatics, Kobe University, Japan

Abstract— With the development of IoT techniques, it become easier to collect users' action data. By analyzing and using those data, consumers and producers will mutually exchange their intelligence and better customize product development processes. This study examines a proposed system using sensor shoes with several sensor devices embedded in the insoles, collecting action data of users, extracting their action features, and then issuing some advice to help users train more efficiently. As described herein, a service model uses a backpropagation (BP) network to distinguish users' actions and to extract their action features using using Self-organization Map from the presented sensing data. Finally, we review their performance via experimentation.

Keywords: Neural network, Self-organizing map, Service Design

I. INTRODUCTION

A. Trends in service research

The service industry's position in the Japanese economy is increasing every year. In the Japanese economy, service industries account for increasing importance and an increasing share of GDP [1]. Various studies examine related domestic issues of innovation and service industry productivity improvement. From 2000, from studies with themes such as individual services, system design for service model construction, sports science and health sciences have gradually gained in value [2]–[3].

Richer lives require an abundance of products and services. With the popularization of information and communication technology (ICT), it has become much easier to obtain more and more various information than ever before. Moreover, consumers tend to choose products or services that are suitable for themselves more freely than before [4], which can raise the question of how to produce the most suitable products or services for consumers with different lifestyles. To ascertain consumers' changing needs, accumulating consumers' historical data showing their motivations and then exploiting needs from the historical data and providing services are expected to be necessary.

B. Development of IoT Technology

The phrase "Internet of Things, "(IoT) was first coined by Kevin Ashton in 1999 in the context of supply chain management [5]. Subsequently with the popularization of cloud computing, the phrase "IoT" tends to represent an environment in which uses cloud computing technology is used with sensor devices [6]. Earlier, researchers had been aware of the concept of "connecting things together". Mark

Weiser also presented a smart environment for people's daily life through this concept [7]. Computers have become much more widely used since then, and have become useful anywhere and anytime. Embedding computer technology seamlessly in daily life, defined as ubiquitous computing, was undertaken and examined in various studies simultaneously. As a result, the number of connected devices exceeded the world population in 2011[8]. Although the definition of "things" has changed with the development of technology, the target of developing a computer that is independent from human beings with the capability of sense information and decision-making function will not change.

C. Utilization of IoT technology

Along with the progression of IoT technology studies, an increasing number of connected devices developed in a larger resources space, it is possible to improve human intelligence, which can also help in improving capabilities [9]. In recent years, development of IoT Technology makes it possible to acquire users' personal data. To customize products using personal data collected by IoT devices, product development has become a co-creation process involving producers and consumers. Anthropometric measurements, used in sports medicine and health science domains, have become used increasingly for applications including the development of products presumed to supposed to support consumers' daily lives [10]–[11]. Regarding studies of IoT technology utilization, the Haptic Design project [12], with the theme of "body experience in the first person," specifically examined improvement of user happiness by sharing or expanding tactus information, or by designing an immersive environment for consumers [13]. Such studies devote attention to producing much better product experiences, but a great deal of discussions arises about the related data. Because IoT devices can also collect data of different kinds, a more important task is how to process data correctly and efficiently [14]. Collecting as much data as possible is important, but little or no improvement can come from merely collecting a huge amount of data without analyzing or using it.

This study specifically examines design of a support service for daily life with consideration of utilization of the sensor data collected from a user.

D. Sensor Shoe

In this study, a pair of sensor shoes [15] is used to collect a user's daily data. Shoes are necessities of life. They can accumulate user action data easily. Moreover, it is possible for a user to spend daily life unconscious of sensors

embedded in an insole. Such sensors are likely to reveal a user's demands and difficulties that a user might not be aware of in daily life. Embedded in an insole of a shoe could be an acceleration sensor, an angular velocity sensor, a geomagnetism sensor, and pressure sensors at three positions of the foot, along with a battery. Every sensor connects with a smartphone and sends sensor information by Bluetooth. Through the application algorithm installed in the smartphone, the user's acceleration, angular velocity, pressure at three positions, and other parameters are calculated and are then shown in the interface.

This paper is organized as follows: the next section explains objective and proposed service. In Section III, methods to implement proposed model are presented. In Section VI, computational experiments are described and results are demonstrated. Finally, conclusions and future work are discussed.

II. OBJECTIVE AND PROPOSAL

A. Objective

This research is aimed at the design of a service that gives advice for training by distinguishing a user's actions and extracting the actions features based on the user's action history data collected by sensor shoes.

B. Proposed service

A service related to the design provides value for a user's life by analyzing and using various information collected from sensor shoes during daily life. The services are divisible into two types according to the data usage. One accumulates only individual data and extracts useful information. The other shares data with other users to provide a reference.

Considering a service that records and collects action history data, a user who trains with sensor shoes has action features and difficulties that can be improved by comparison to previously set goal data. A user can then train much more efficiently at places not restricted to a gym. This report specifically describes training support service implementation.

Because the movement state is not the same every time a user trains, recording various movement states is important for further analysis. The flow of the proposed service is the following.

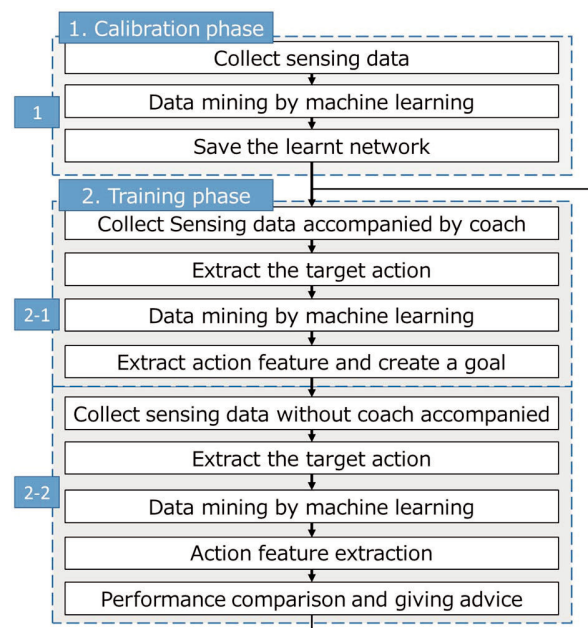
- Goal setting: A user trains with a coach, records the sensing data, visualizes the action features by data mining, then sets a goal.
- Training: A user trains without a coach, accumulates sensing data, then extracts and visualizes the action features by data mining.
- Advising: The system compares results of training scores and advises a user to achieve the goal data.
- Goal data updating: A user who achieves the goal trains again with coach to update the goal data.

C. Proposed model

A model of training support service is depicted in Fig. 1, which consists of a calibration phase and training phase.

- Step 1: Calibration phase. Record various action data of a user as training data and train a network for distinguishing actions.
- Step 2-1: Training phase. Train with coach to set the goal data. A user trains according to the coach's instructions. Use the trained network in Step 1 to extract the target action data collected by sensor shoes from sensing data of each training bout. Use machine learning for data mining. Extract action features. Visualize goal data.
- Step 2-2: Train without coach while providing services. In this step, a user trains without a coach. Accumulate action data of each training bout and visualize action features. Compare the action features of each training bout and provide advice to approach the goal efficiently.

Figure 1. Proposed model.



III. METHOD

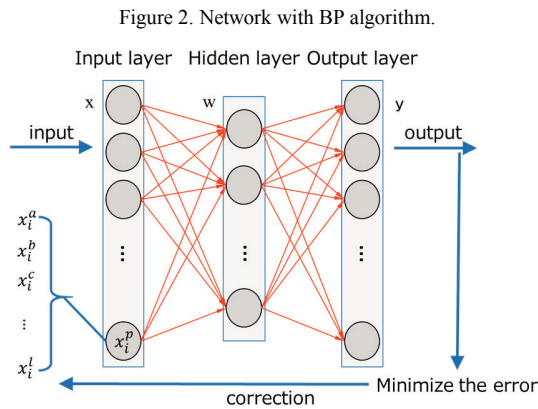
Machine learning will be used to implement data mining. In machine learning, sample datasets are input from sensors, databases, etc. and are analyzed. From those data, useful rules, knowledge representation, judgment criteria are extracted to develop algorithms [16].

Considering that the proposed model has a characteristic by which the action varies from person to person, distinguishing the actions of individual users correctly is necessary. This research uses supervised learning at the calibration phase. A user performs actions that may be done in training with the consciousness of "collect correct training data" to collect training data for training the network. In addition, as support for training, it is important to classify details of a user's sensor data, analyze them, and record action features. Therefore, this research uses unsupervised learning for feature extraction in step 2. The network automatically records and learns various action features when the user trains.

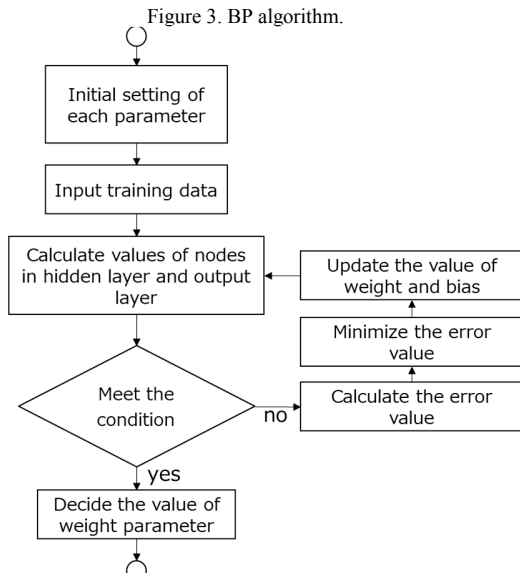
A. Action distinction using Neural Network

In step 1, this paper specifically examines jogging training. Sensing data of actions of five types will be collected for standing, sitting, walking, jumping and running. Then supervised learning will be used to train the network. Subsequently, the actions of new sensing data are distinguished based on the trained network. Target action data are extracted and used for the next step.

A neural network with a backpropagation (BP) algorithm will be used to distinguish actions. BP is a neural network algorithm that corrects weights and biases by comparing training data and output [17]–[18]. As Fig. 2 shows, the network has an input layer, a hidden layer and an output layer. Neurons exist in each layer. The entire network is configured by connecting neurons. Weight w is calculated based on the importance of the input signal and that of the error signal. Then it is updated in the hidden layer.



Common BP algorithm can be described as shown below.



- A) Initial setting of each parameter.
- B) Input training data.
- C) Calculate the values of nodes in the hidden layer and output layer.
- D) Verify the gradient of the error to judge if learning has ended. If the condition is satisfied, then the value

of the weight vector is determined and learning is ended. Otherwise the parameters are updated by backward propagation.

- E) Calculate the error value
- F) Minimize the value of error by gradient descent
- G) Update the values of weight and bias; return to (C).

In the input layer, 10-dimensional data are used for each neuron: x_t^p ($p = a, b, c, \dots, h, k, l$).

t : point in time

a : left/right foot

b, c, d : acceleration axis x - z .

e, f, g : angular velocity axis x - z .

h, k, l : pressures of heel, big toe, and little toe positions.

Weights between the input layer, hidden layer, and output layer are defined as w_{ij}, w_{jk} . Outputs of the hidden layer and output layer are defined as H_j and O_k .

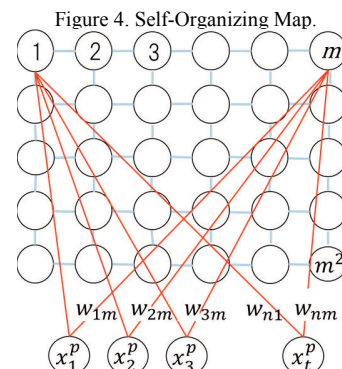
In the output layer, output is defined as shown below.

$$y_i = \begin{matrix} (\text{stand}) & [1 & 0 & 0 & 0 & 0] \\ (\text{walk}) & [0 & 1 & 0 & 0 & 0] \\ (\text{run}) & [0 & 0 & 1 & 0 & 0] \\ (\text{sit}) & [0 & 0 & 0 & 1 & 0] \\ (\text{jump}) & [0 & 0 & 0 & 0 & 1] \end{matrix}$$

A sigmoid function is used as an activation function: $g(x) = \frac{1}{1+e^{-x}}$. Error is calculated as $E = \frac{1}{2} \sum_{k=1}^m (Y_k - O_k)^2$, where Y_k and O_k respectively denote the expected result and output. Gradient descent is used to make E least.

B. Feature extraction using self-organizing map

Self-organizing Map (SOM) will be used for feature extraction to implement goal data setting and advising steps. SOM is a kind of unsupervised learning neural network proposed by Teuvo Kohonen [19] consisting of two layers: an input layer and an output layer. Input layer N_{in} has n -dimensional numerical data. Output layer N_{out} has nodes arranged in n -dimensional space. A weight vector w_{nm} is attached to each node. SOM is trained by updating weight vectors. Figure 4 depicts the construction of a common SOM network.



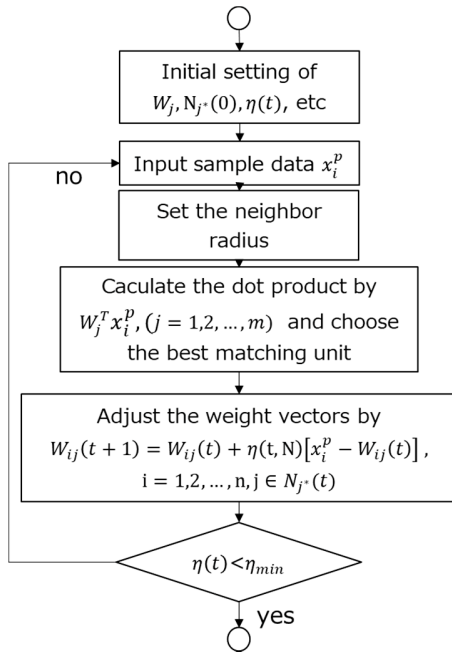
Because the dimensions of the node arrangement of the output layer can be set freely, one can visualize the relation between the data by arranging the nodes in low dimensions and learning complex high dimensional data.

In the input layer, after a sample is selected randomly, its weight and distance between every neuron in the input layer are calculated. The most similar neuron can be chosen as a winner node. Subsequently neuron weights around the winner node will be updated. As described in this paper, a 10-

dimensional data x_t^p as section III-A are also used for each neuron in the input layer.

The algorithm can be described as explained below:

Figure 5. Algorithm of SOM.



- A) Initial setting of each parameter.
- B) Input sample data x_t^p .
- C) Set the neighbor radius N_{j^*} .
- D) Decide the winner node j^* .
- E) Update the weights.
- F) If learning coefficient η arrive the set value, then finish the process; otherwise return to (C).

IV. EXPERIMENTS

A. Action distinction

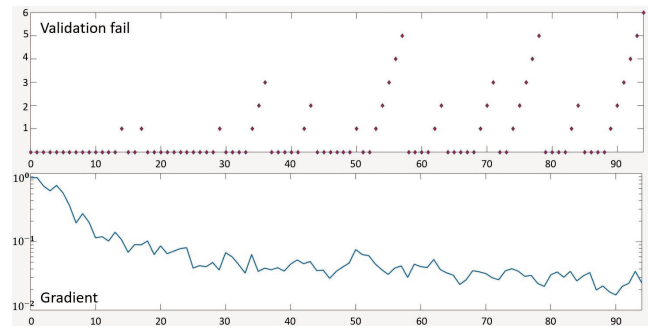
In this section, an experiment is conducted to consider the accuracy of the action distinction method using a BP algorithm. The experimental condition is the following.

The user performs sitting, standing, walking, running, and jumping actions for 2 min each to collect sensing data. The measurement frequency of sensors is 10 times/s, data of both feet are integrated. 2400 sensing data are collected for each action. Then 500 samples are extracted continuously from them randomly. Data extracted in each action are combined to form a matrix comprising 2500 columns and 10 rows.

- Number of input layer nodes: 10
- Number of hidden layer nodes: 100
- Number of output layer nodes: 5

End condition of learning by validation verification: As described in this paper, if the gradient of the error does not change after six consecutive iterations, then learning is terminated to suppress over-learning. Learning was finished in the 94th epoch by validation verification. The gradient change of the error is shown in Fig. 6. The accuracy rate was 74.1%. In this experiment, training data were set at each time point, but because each parameter of action has a period, training data set in a fixed time series cycle are regarded as improving the accuracy.

Figure 6. Validation verification and gradient change of network.



B. Feature extraction

This section describes investigation of the effectiveness of the proposed method. Also, we conduct three experiments to assess action feature analysis by SOM. First, experiment 1 is designed to ascertain whether the user's action can be inferred from the cluster results. Experiment 2 is designed to investigate different states of the same action of one user. Finally, experiment 3 is designed to examine whether differences between the states of the same action of multiple users can be discriminated by cluster results. The conditions of the respective experiments are explained below.

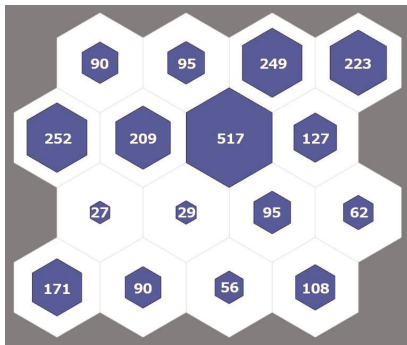
- Experiment 1: The user wears sensor shoes and performs various actions to collect sensing data. About 3 min of walking action data are extracted as training data.
- Experiment 2: The user jogs about 2 min according to the instructions of a coach. Sensing data are collected and used as learning data (coach data). Subsequently, a user jogs about 2 min independently, collects sensing data, and uses the data as learning data (training data).
- Experiment 3: Two users jog the same distance. Collect sensing data and use it as learning data called data A and data B.

1) Experiment for action state inference based on cluster results

Results of clustering based on 2400 action data are obtained using Matlab SOM tools, as shown in Fig. 7. On the cluster map, the neurons are at the center of each cluster. The area of the blue part indicates the number of data assigned to the respective clusters.

Analyses of action features are based on mean values of each sensing data parameter. First, according to the pressure on the heel, the little toe and the big toe's position of each cluster's data, the user's action state is classified as a "foot is landing condition" or "foot is away from the ground condition".

Figure 7. Clustered data.



- Condition of landing

Based on the magnitude of the pressure value at each position, the landing site is classified as “heel” or “toe”. In addition, use of the z acceleration value can imply the weight of landing.

- Condition of foot above from the ground

In this condition, the pressure at each position is almost zero. Therefore, it is better to examine the acceleration

specifically. According to the values of acceleration x and y axes, data are classified into three conditions: “putting down the foot”, “lifting the foot” and “passing the zenith”. Next, while referring to the z-axis acceleration values and the angular velocity, the momentum of walking and other detailed features can be inferred. The average value of each parameter calculated in each cluster is shown in TABLE I: the so-called feature table.

Figure 8. Conditions of foot away from the ground.

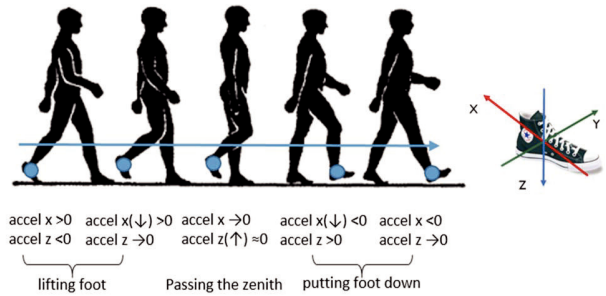
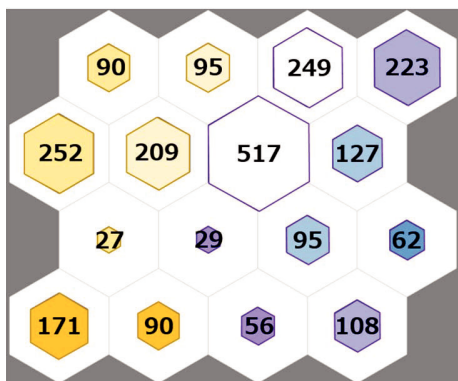


TABLE I. FEATURE TABLE

Cluster parameter	1	2	3	4	5R	6	7	8	9R	10R	11	12	13L	14L	15	16
number	171	90	56	108	27	29	95	62	252	209	517	127	90	95	249	223
condition	1	1	0	0	1	0	0	0	1	1	1	0	1	1	0	0
accel x	L-0.77, R-0-1.97	-0.48	41.59	2.84	-0.15	33.51	-12.53	-82.74	-0.56	-0.15	-1.02	-22.79	-0.84	-0.18	8.37	5.11
accel y	L-1.91, R7.24	0.76	L18.51, R-11	1.81	0.04	-1.68	2.43	4.89	0.65	0.33	0.8	1.04	0.26	1.24	-0.16	2.81
accel z	24.89	16.73	42.13	64.83	17.33	-96.86	23.38	28.34	19.75	19.83	16	41.56	21.68	21.2	9.81	25.46
gyro x	L-6.42, R2.62	-0.27	1.85	-5.18	0.26	-11.07	1.14	L-3.57, R7.72	-0.89	-0.11	-0.63	3.72	-1.2	-2.73	L10.02, R-5.85	L10.58, R-6.78
gyro y	20.44	1.88	64.07	62.6	0.89	33.17	33.49	L-2.77, R-5.41	5.98	4.69	0.14	-41.61	9.58	10.88	-26.03	44.12
gyro z	L4.42, R-19.31	-0.69	L-30.5, R7.19	-2.82	-1.07	-3.48	-3.73	L4.17, R-5.94	-1.04	-1.01	-0.62	7.26	-0.93	-1.54	L6.94, R0.54	L4.64, R-2.98
heel	104.61	54.32	0	0	60.7	9.93	0.38	0	0.21	1.69	1.33	0	0	0.32	0	0
little toe	0	0	0	0	68.37	0	1.54	0	107.81	75.84	1.56	0	46.31	5.64	0	0
big toe	0	0	0	0	1.52	0	1.96	0	42.08	9.63	3	0	99.89	75.15	0	0

In the feature table, if data of only one foot are classified into a cluster, then an L or R is appended after the cluster number in the first line. The state shown in the second line is described with 1 and 0, signifying “condition of landing” and “condition of foot away from the ground”.

Figure 9. Colored cluster result of Experiment 1.



Based on the feature table, the coloring of cluster results is shown in Fig. 9. Yellow represents the landing condition. The darker cluster shows that the heel pressure is greater. The lighter cluster shows that the toe pressure is greater. Blue represents “putting down the foot”. Also, lighter colour show

smaller x direction acceleration. Purple shows “lifting of the foot”, with lighter colours representing smaller x direction acceleration. The other conditions are shown in transparent color. Features of walking can be investigated. Similar conditions are colored similarly based on the table. Results shows that clusters with similar conditions are assigned in the near place. Therefore, one can use SOM to analyse a user’s action features.

2) Experiments to assess action states from one user

The colored cluster results of 1412 coach data and 1340 training data using SOM are shown in Fig. 10. Regarding training data, the data classified as being in the heel on the ground condition account for 4.75% of the total number of data classified in a condition of landing. However, regarding the coach data, the data classified in the heel on the ground condition account for 10.15% of the total number of data classified in the condition of landing. Therefore, one can speculate that a user who trains according to a certain style will often land on the toes. When the foot is away from the ground, the pressure and z acceleration values are lager when a user trains with a coach. Therefore, if a user trains independently, the momentum of the action might be smaller. Therefore, some advice can be proposed: “Land by the heel more if you want to achieve the goal efficiently.

3) Experiment on comparing action states of multiple users

The colored cluster results of 1371 data A and 1177 data B using SOM are shown in Fig. 11. Regarding the condition of the foot landing, it can be speculated that user A lands more often than user B because user A occupies 62.71% and user B occupies 81.44% of the total number of data. Furthermore, the ratio of the number of data for lifting the foot when the foot is away from the ground is 63.86% for A and 51.73% for B. It can be speculated that user A takes more time for lifting of the foot than user B. Consequently, SOM can suggest different features of the same action of multiple users.

Figure 10. Colored cluster results of coach data (left) and training data (right).

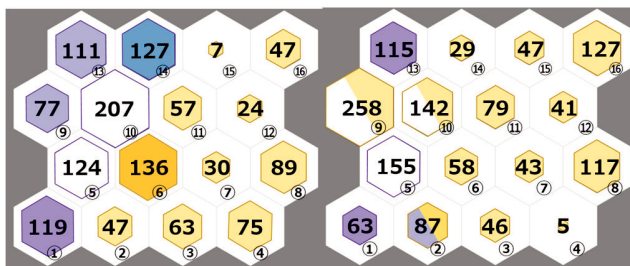
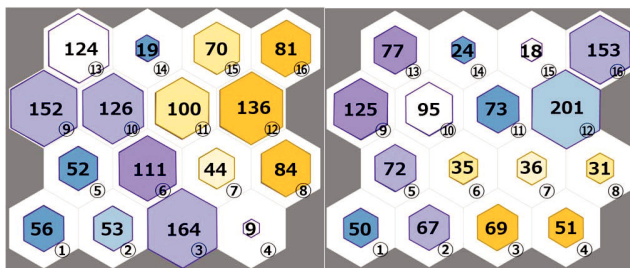


Figure 11. Colored cluster results of data A (left) and data B (right).



V. CONCLUSIONS

In action distinction, although one can distinguish actions by sensing data from one user using the BP algorithm, the accuracy is still low. Therefore, it is necessary to improve the data processing or algorithm itself. In feature analysis, one can infer a user's action state from cluster results obtained using SOM. Then one can investigate each cluster's feature. Moreover, one can compare different states of the same action of one user from extracted features, making it possible to propose advice to support efficient training. However, the cluster position is fluid in every learning result. Therefore, future studies must investigate rules of assignment.

REFERENCES

- [1] Ministry of Economy, Trade and Industry of Japan, Trade white paper of 2015. 2016.
- [2] T. Takeshi, K. Naito, K. Ueda, Service Research Strategy toward Value Co-Creation, Information Processing Society of Japan, Vol. 50, pp. 1539-1548, 2008.
- [3] T. Magome, Development in Sports Analysis and Sports Medicine domain using IoT sensor Technology, Informatics Society Magazine, No. 46, pp. 112-116 2018 (translated from Japanese).
- [4] K. Naito, Service Engineering, University of Tokyo Press, 2009.
- [5] K. Ashton. That "internet of things" things. *RFid*, Vol. 94, 2009.
- [6] ITU. Internet of things global standard initiative. 2015.
- [7] R. Gold, M Weiser. The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems*, 1999.
- [8] R. Buyya J. Gubbi. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, Vol. 29, pp. 1645-1660, 2013.
- [9] Y. Rogers. Moving on from Weiser's vision of clam computing: engaging ubicomp experiences. *Ubiquitous Computing*, 2006.
- [10] IoTechNews, A wearable device improves your eating habits (translated from Japanese), iotechnews.com/2015/01/14/bitbite/.
- [11] U-NOTE, An electric toothbrush with embedded Bluetooth.(translated from Japanese), u-note.me/note/47499535.
- [12] K. Minamizawa, Haptic design project, hapticdesign.org.
- [13] R. Yokoyama, An extraordinary experience produced by the combination of vision, auditory sense and tactus, hapticdesign.org/designer/file016_Ryo_Yokoyama.
- [14] M. Hassanalierege, A. Page, Health monitoring and management using internet-of-things (IoT) sensing with cloud-based processing: Opportunities and challenges. *IEEE International Conference on Services Computing*, Vol.47, pp. 285-292, 2015.
- [15] Fujitsu Limited–interactive shoe hub-android API. *Fujitsu Confidential*, 2016.
- [16] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [17] P. Flach. *Machine learning-The art and science of algorithms that make sense of data-*. 1997.
- [18] G. Ian, B. Yoshua, C. Arraon. *Deep learning*. MIT Press, 2016.
- [19] T. Kohonen, *Self-Organizing Maps*. Springer Series in Information Sciences, 2001.

A study on support method of consulting service using text mining

-Application to real problem-

Ruriko Watanabe
Kobe University
Japan

watanabe@kaede.cs.kobe-u.ac.jp

Nobutada Fujii
Kobe University
Japan

nfujii@phoenix.kobe-u.ac.jp

Daisuke Kokuryo
Kobe University
Japan

watanabe@kaede.cs.kobe-u.ac.jp

Toshiya Kaihara
Kobe University
Japan
kaihara@kobe-u.ac.jp

Yoichi Abe
F&M Co., Ltd.
Japan
yoichi_abe@fmltd.co.jp

Ryoko Santo
The New Industry Research
Organization
Japan
santo@niro.or.jp

Abstract— This study was conducted to produce a supporting method for consulting service companies so that the companies can respond to client demands irrespective of the consultant expertise. Occurrence of future problems in client companies is predicted using text mining with data taken from a consulting company. To do so, correspondence analysis and Data Envelopment Analysis (DEA) discriminant analysis are used. As described herein, the proposed method is improved using Intermediate Distribution Frame (IDF) values and standardization for the application in a real problem. For computer experiments to verify the effectiveness of the proposed method, prediction is made about cancellation problems. The prediction is compared to predictions produced by the consultant.

Keywords—Correspondence analysis, DEA discriminant analysis, Text mining

I. INTRODUCTION

In recent years, with emphasis on the activation of small and medium-sized enterprises [1], support systems of consulting for small and medium-sized enterprises are regarded as important [2]. Such support for small and medium-sized enterprises is expected to solve problems that are difficult to address within their company. Although consulting companies can respond to widely various needs for management consultation, they have a difficulty that leading in a special field is not always sufficient [3]. In addition, service suggestions and client companies' problem detection depend on the experience and intuition of each consultant. An auxiliary system is necessary to provide services with stable quality that is independent of a consultant's ability. This study was conducted to realize consulting services that are independent of a consultant's expertise. As a first step in constructing an auxiliary system, this study proposes a method to predict the occurrence of future problems in client companies using text mining with data representing communication about consultation-related matters.

In earlier studies [4], a discriminant was made using real-scale data to address cancellation difficulties. The effectiveness of the proposed method was confirmed. Those study findings confirmed that the difference between the numbers of cancelling companies and continuing companies affects discriminant analysis. By extending the objective function of Data Envelopment Analysis (DEA) discriminant analysis, it was possible to suppress the influence of the differences in the numbers of companies.

As described in this paper, to resolve difficulties imposed by the real scale problem, the method is extended to solve the following three problems.

- The text of the prediction data amount is smaller than that of the training data. The overlapping of words is slight. The method is extended to the use of words that appear in only one group.
- Although factor words were extracted using correlation between the company and the word, with the increase in the number of companies, a problem was found by which a word unique to a company with a large amount of text data was also extracted. The method was extended using Intermediate Distribution Frame (IDF) values to extract more appropriate words.
- Normalization was introduced to normalize the word distribution.

To verify the effectiveness of the proposed method, prediction is made about cancellation problems. Results are compared with prediction results obtained from a consultant.

II. METHOD

A. Research Subject

By analyzing text data accumulated in consulting companies, possible problems in client companies are predicted from this study. The text data to be used were recorded through various methods, such as the interaction at the time of a visit between the consultant and client company, questions from the client company provided by e-mail, and correspondence with a call center. Examples of such data are presented below.

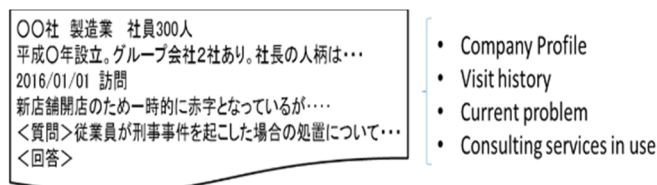


Fig. 1. Recorded text data.

As presented in Fig. 1, the text data consist of the company profile, visitation history, difficulties currently confronted by the client companies, and the consulting services in use. Some texts appear to describe contents that might trigger

future problems. For this study, these text data were analyzed using a computer to support consulting services. The proposed method supports consultants who must address difficulties encountered in widely various fields, allowing consultants with little experience to be aware of difficulties that can occur in the client companies by predicting them from the accumulated text data.

B. Overview of Proposed Method

This section presents an outline of the proposed method. First, text data are classified based on the problem occurrence. A discriminant formula is created using a text mining method. Phrases are extracted as factors by extracting the correspondences for each group from among many different words and phrases, and using the extracted phrases as variables of a DEA discriminant analysis to create a discriminant. To verify the effectiveness of the obtained discriminant formula, newly categorized text data are used to judge the presence or absence of a difficulty. A flow of the proposed method is presented in Fig. 2.

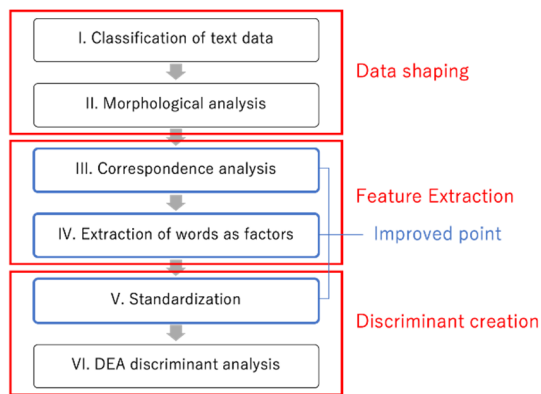


Fig. 2. Flow of proposed method.

The proposed method is conducted through the following six steps. These steps are presented in Fig. 2. As described in this paper, the method was improved by the introduction of IDF values in III, the use of words that appear in only one group in IV, and the standardization in V. Detailed descriptions of the respective steps are presented in the following sections.

- | | |
|-------------------------------|--------|
| I. Classify the text data | (II.C) |
| II. Morphological analysis | (II.D) |
| III. Correspondence analysis | (II.E) |
| IV. Extract words as factors | (II.F) |
| V. Standardization | (II.G) |
| VI. DEA discriminant analysis | (II.H) |

C. Classify the Text Data

This study used text data related to communication with the client companies accumulated by a consulting company. The text data are categorized according to the presence or absence of problem occurrence when considering the time series of problem detection. To predict problem occurrence, text data with direct content on the problem are deleted. In addition, past text data for a certain period of time since the problem occurred are applied because the proposed method tries to predict the occurrence of a problem as early as possible by not applying a direct description of the problem in the text data.

D. Morphological Analysis

A morpheme, the smallest character string, becomes meaningless when it is decomposed further. Decomposing sentences into morphemes and specifying their parts of speech is called morphological analysis. This study used MeCab [5], a morphological analyzer developed by Kudo as a general-purpose analyzer that is independent of dictionaries or text data. It is faster than other analyzers. Words and phrases of the following items, which should be considered noise in analyses, were omitted from the analyzed morphemes.

- Name of consulting service
- Fixed phrase
- Incoherent words

E. Correspondence Analysis

1) Correspondence analysis overview

Correspondence analysis, proposed in the 1960s by French researcher Jean-Paul Ben-zeccri, is a method to compress information in the rows and columns of a data table into a few components [6]. In this study, words are included in line items (sample). Company names are included in column items (category). Two-dimensional data of the appearance count t_{ij} of word j of company i are targeted. Table 1 presents two-dimensional data: a_i is the sample score; b_i is the category score. Each is set as a variable.

TABLE I. DATA FOR CORRESPONDENCE ANALYSIS

	Word	AAA	BBB	...	KKK
Company		a_1	a_2	...	a_K
A Co.	b_1	t_{11}	t_{12}	...	t_{1K}
B Co.	b_2	t_{21}	t_{22}	...	t_{2K}
⋮	⋮	⋮	⋮	⋮	⋮
N Co.	b_N	t_{N1}	t_{N2}	...	t_{NK}

Variables are set for samples and categories. Calculations are done to maximize inter-variable correlation. By mapping sample scores and category scores corresponding to each axis on the scatter diagram, correspondence between these variables can be visualized. Characteristic companies and words in the target data appear as they move away from the origin of the scatter plot. General companies and words in the target data appear near the origin. For this study, correspondence analysis is conducted for each group divided by the presence/absence of fraud problem detection. Also, attention is devoted to the word close to the origin. Words which become factors of the respective groups are extracted using the method shown in the next section.

2) Calculating the word factor strength

The results of correspondence analysis showed, considering all dimensions, distance d_{iG} between word i . The origin in each group (G_1, G_2) is calculated using Eq. (1). Here, D_G represents the total number of dimensions, x_{ijG} is the sample score of word i in dimension j , and C_{jG} denotes the contribution of dimension j .

$$d_{iG} = \sqrt{\frac{\sum_{j=1}^{D_G} \{x_{ijG} * C_{jG}\}^2}{D_G}} \quad (1)$$

Weighting is performed by multiplying the distance from the origin by the IDF value to extract more commonly used words. The IDF value, called reverse document frequency, is a high value if the word is rare between documents, and a low value if the word appears frequently in many sentences. IDF values are calculated using Eq. (2).

$$idf_i = \log \frac{N}{N'_i} \quad (2)$$

In that equation, N denotes the total number of documents; N'_i represents the number of documents including word i .

F. Extract Words to be Factored

1) Extracting words appearing in both groups

For each word, the d value is updated by the operations of Eq. (3) and (4). For each group, a fixed number of words with small e values are extracted.

when $G_1 < G_2$

$$\begin{cases} e_{iG_1} = d_{iG_1} * idf_i + M - d_{iG_2} * idf_i \\ e_{iG_2} = \infty \end{cases} \quad (3)$$

when $G_1 > G_2$

$$\begin{cases} e_{iG_2} = d_{iG_2} * idf_i + M - d_{iG_1} * idf_i \\ e_{iG_1} = \infty \end{cases} \quad (4)$$

2) Extracting words appearing in only one group

In each group, the following heuristic rules are used to extract the words as factors while also considering the words and phrases that appear in only one group.

- I. From the words appearing in both groups, select the word with the strongest factor: The word with the highest e_{iG} value.
- II. Words that appear in only one group with distance less than the selected words are extracted in order from a word with a smaller distance. The process ends when the set number of extracted words is reached. Exit when the set number of extraction words is reached.
- III. Extract the word selected in I. Go to I if it is less than the set number of extraction words. Exit when the set number of extraction words is reached.

G. Standardization

A discriminant is created using the words extracted in section F. Before performing discriminant analysis, standardization is applied to unify the word scale. The standardization is calculable by Eq. (5) when the average frequency of occurrence of a word is M and the standard deviation is s . The average of each word is 0. The variance is 1.

$$z = \frac{x - M}{s} \quad (5)$$

Results show that, by standardization, a word having a number of appearances of 0 is also assigned a value. It can be determined.

H. DEA Discriminant Analysis

The DEA discriminant analysis method is a discrimination model proposed by Sueyoshi [7]. A DEA discriminant analysis is conducted in two stages. In the first stage, the data are classified into two groups: properly discriminated data and data that cannot be discriminated or cannot be discriminated easily. In the second stage, improving the accuracy of the discrimination, a discriminant analysis is applied to data that are difficult or unable to detect in the first stage. A model of the applied DEA discriminant analysis is shown below. Table II presents variables used for DEA discriminant analysis.

TABLE II. VARIABLE USING DEA DISCRIMINANT ANALYSIS

Decision variables	
G	Group
i	Extracted word
j	Company
z_{ij}	No. of word i occurrences in company j text data
η	Width of overlap
Dependent variables	
λ	Discrimination coefficient
d	Discrimination boundary
S_{ij}^+, S_{ij}^-	Slack variable

The objective function of Eq. (11) is designed to minimize false discrimination. When λ_i^*, d^* are the optimal solutions obtained in stage 1, companies are classified into five categories according to the following discrimination criteria: R_0, R_1, R_2, C_1 , and C_2 .

Stage 1

$$\begin{aligned} \min \quad & \sum_{j \in G_1} S_{1j}^+ + \sum_{j \in G_2} S_{2j}^- \\ \text{s.t.} \quad & \sum_{i=1}^k \lambda_i z_{ij} + S_{1j}^+ - S_{1j}^- = d + \eta \quad (j \in G_1) \\ & \sum_{i=1}^k (\lambda_i^+ - \lambda_i z_{ij}) + S_{2j}^+ - S_{2j}^- = d \quad (j \in G_2) \\ & \sum_{i=1}^k |\lambda_i| = 1 \\ & S_{1j}^+, S_{1j}^-, S_{2j}^+, S_{2j}^- \geq 0 \end{aligned} \quad (6)$$

The objective function of Eq. (6) is designed to minimize false discrimination. When λ_i^*, d^* are the optimal solutions obtained in stage 1, the companies are classified into five categories according to the following discrimination criteria: R_0, R_1, R_2, C_1 , and C_2 .

$$R_1 = \left\{ j \in G \mid \sum_{i=1}^k \lambda_i^* z_{ij} \geq d^* + \eta \right\} \quad (7)$$

$$R_0 = \left\{ j \in G \mid d^* + \eta > \sum_{i=1}^k \lambda_i^* z_{ij} > d^* \right\} \quad (8)$$

$$R_2 = \left\{ j \in G \mid \sum_{i=1}^k \lambda_i^* z_{ij} \leq d^* \right\} \quad (9)$$

$$C_1 = \{j \in R_1 \mid j \in G_1\} \quad (10)$$

$$C_2 = \{j \in R_2 \mid j \in G_2\} \quad (11)$$

Therefore, C_1 and C_2 are classified correctly. Here, $G_1 \cap R_2, G_2 \cap R_1$ is a dataset that was misjudged. Set R_0 is made up of data in the overlapped region.

In stage 2, misidentified data and data existing in the overlapping area are handled. Variable c is a new discrimination boundary existing between d^* and $d^* + \eta$.

Stage 2

$$\begin{aligned} \min \quad & \sum_{j \in G_1 \cap (R_0 \cup R_2)} S_{1j}^+ + \sum_{j \in G_2 \cap (R_0 \cup R_1)} S_{2j}^- \\ \text{s.t.} \quad & \sum_{i=1}^k \lambda_i z_{ij} + S_{1j}^+ - S_{1j}^- \\ & = c \quad (j \in G_1 \cap (R_0 \cup R_2)) \\ & \sum_{i=1}^k \lambda_i z_{ij} + S_{2j}^+ - S_{2j}^- = c \quad (j \in G_2 \cap (R_0 \cup R_1)) \\ & \sum_{i=1}^k \lambda_i z_{ij} \geq d^* + \eta \quad (j \in C_1) \\ & \sum_{i=1}^k \lambda_i z_{ij} \leq d^* \quad (j \in C_2) \\ & \sum_{i=1}^k |\lambda_i| = 1 \\ & S_{1j}^+, S_{1j}^-, S_{2j}^+, S_{2j}^- \geq 0 \end{aligned} \quad (12)$$

Data correctly determined in stage 1 are controlled using a constraint expression to ensure the results of stage 1. In the objective function, the slack variables of data correctly discriminated in stage 1 are excluded. Therefore, the sum of the slack variables that occurs when the data are erroneously discriminated in stage 1 and also discriminated erroneously in stage 2 is minimized. In addition, the discrimination boundary value c is set as between d and $d + \eta$. It is

possible to discriminate data existing in the overlapped region. If the optimal solution of stage 2 is c^*, λ_i^* , then data are judged according to the following criteria. In Eq. (13), company j is judged to belong to G_1 . In the case of Eq. (14), company j is determined to belong to G_2 .

$$\sum_{i=1}^k \lambda_i^* z_{ij} \geq c^* \quad (13)$$

$$\sum_{i=1}^k \lambda_i^* z_{ij} < c^* \quad (14)$$

III. COMPUTER EXPERIMENTS

A. Experiment Overview

In computer experiments, actual data are analyzed using the proposed method as a target of the problem of cancellation. Specifically, a discriminant that predicts client consulting service cancellation is created. To solve a real-scale problem, the method was extended to solve the following three points. The validity of the extended method was verified.

- Using words that appear in only one group.
- IDF values
- Standardization

To verify the effectiveness of the proposed method, prediction is conducted for a cancellation problem. Results are compared with the prediction results from a consultant.

B. Target Data

Describe data when predicting a company for which n month d year is the update month. Figure 3 shows data (learning data) for discriminant preparation. Data of the company's prior nine months were used. However, the earlier three months were deleted from the renewal month because the study is aimed at making a cancellation projection before the consultant approaches the cancellation, which requires a three month period.

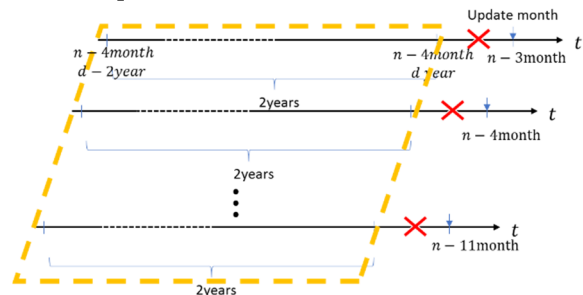


Fig. 3. Learning data

Fig. 4 shows data (prediction data) for target month prediction. It accomplishes prediction using text data of the company which is the target update month. Similarly to data for creation, we delete the data of the three months before the update month.

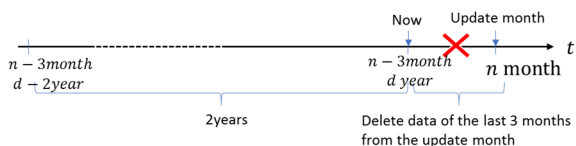


Fig. 4. Prediction data

C. Common Experiment Conditions

The experiment conditions are shown below. In this study, although machine learning is not used, data used for discriminant creation are designated as learning data. Data used for verifying the discriminant are designated as prediction data. Table 4 expresses the condition.

Experiment environment

TABLE III. EXPERIMENT CONDITIONS

	Cancellation Group: G_1	Continuation Group: G_2
Learning data	315	4048
Prediction data	110	451
No. of extracted words	500	
Width of overlap	0.5	

D. Effectiveness of method improvement

1) Learning data

Table IV presents the discrimination rate of learning data and the number of companies that can not discriminate. Here, the undeterminable company is a company with words used in the discriminant which do not appear in the text data and which can not be discriminated in either group. In other words, it is not the number of misclassified companies.

TABLE IV. DISCRIMINANT RATIO OF LEARNING DATA

Method before improvement		
	Cancellation	Continuation
Discriminant ratio	96.8	97.6
No. of undetermined companies	253	3715
Improved method		
	Cancellation	Continuation
Discriminant ratio	99.0	99.9
No. of undetermined companies	0	0

Both methods have discrimination rates close to 100% for both groups. The existing method showed about half the number of undetermined companies existed. However, using the improved method, the number of unidentifiable companies became zero. This result is regarded as attributable to the fact that “the phrase does not appear” also contributes to the determination factor because the words are standardized and the occurrence frequency is assigned a value of 0.

2) Prediction data

Table V shows the discrimination rate of prediction data and the number of companies that can not be discriminated.

TABLE V. DISCRIMINANT RATIO OF PREDICTION DATA

Method before improvement		
	Cancellation	Continuation
Discriminant ratio	0.9	100
No. of undetermined companies	109	448
Improved method		
	Cancellation	Continuation
Discriminant ratio	72.0	66.7
No. of undetermined companies	60	271

In the existing method, the discrimination rate of the continuation group is 100%, but the discrimination rate of the cancellation group is 0.9%. Predicting the company to be cancelled is not possible. In addition, most companies have become undetermined companies. However, using the improved method, the termination rate was 72%. The continuation rate was 66.7%. It was easier to create discriminants without bias among groups than by existing methods. Although the undetermined companies were about half of all companies, they were fewer than under the existing method. The effectiveness of the improved method was confirmed from the above.

E. Comparison with consultants' forecasted continuation

Fig. 5 and Fig. 6 respectively present the forecasted rates of updating by consultants and the number of firms in the results that were forecast using this method.

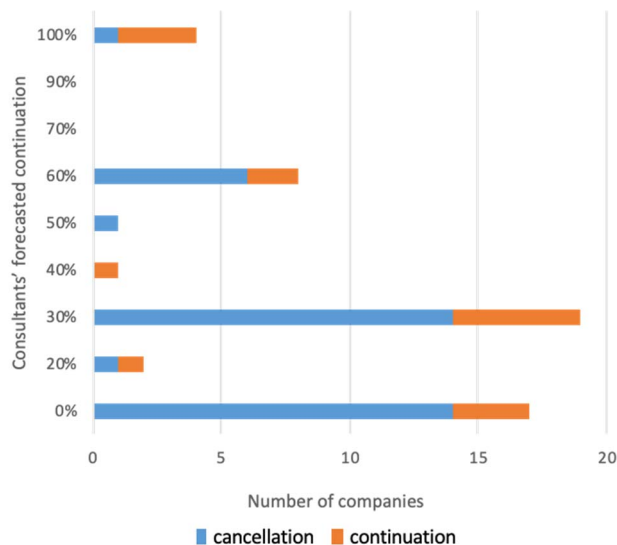


Fig. 5. Cancellation group.

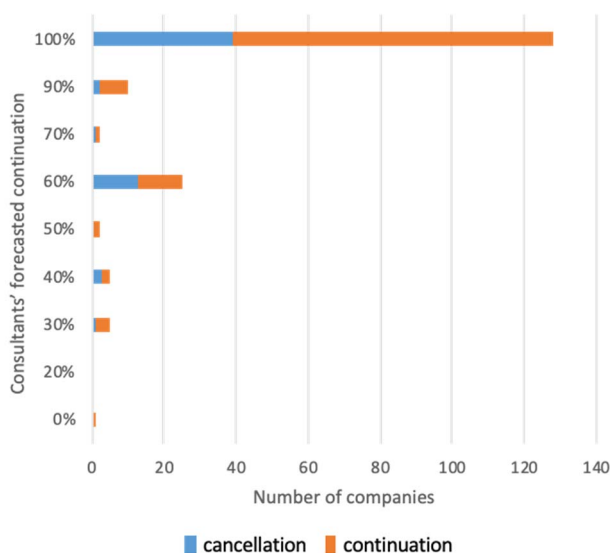


Fig. 6. Continuation group.

The vertical axis presents the rate at which the consultant predicts that the target company will continue consulting services. The horizontal axis shows the number of companies.

The support system correctly predicted many companies for which consultants' forecasts were correct. The results confirmed that it is difficult for the support system to determine cancellation for a company for which the consultant predicted an update as likely to be high but for which cancellation occurred. Results suggest that the support system also captures characteristics of the company when the consultant predicts.

IV. CONCLUSION

This study was aimed at producing a method for supporting consulting service companies so that companies can respond to client demand irrespective of the company expertise. Occurrence of future difficulties in client companies is predicted using text mining with data taken from a consulting company. Also, correspondence analysis and DEA discriminant analysis are used. As described in this paper, the method has been improved to apply real problems in three respects: using words that appear in only one group, an IDF value, and standardization. In computer experiments conducted to verify the effectiveness of the proposed method, prediction is made about cancellation problems. The results are compared with the prediction results achieved by the consultant. The obtained results are summarized below.

- The improved method reduced the number of companies for which results can not be determined.
- The improved method made it possible to create discriminants without bias among the groups, yielding respective discrimination rates of 72% and 66.7% in the cancellation and continuation groups.
- Comparison with forecast results produced by consultants suggests that the support system also captures characteristics of the company to be referred to when the consultant predicts.

This study examined a method to support state recognition in client companies of consultants, but a support method for

the consultant's judgment system is also necessary to construct supplementary systems.

REFERENCES

- [1] Small and Medium Enterprise Charter, 2010.
- [2] http://www.chusho.meti.go.jp/sme_english/index.html
- [3] White Paper on Small and Medium Enterprises in Japan, 2017.
- [4] Ruriko Watanabe, Nobutada Fujii, Daisuke Kokuryo, Syunya Kaihara, Yoichi Abe, Yoshinobu Onishi, Ryoko Santo: A Study of Supporting Method for Consulting Service using customer information, International Conference on Serviceology, November 2018.
- [5] <http://taku910.github.io/mecab/> (in Japanese)
- [6] Shinichiro Shida, Tadahiko Maeda, Makoto Yamazaki, Statistical Input for Language Research Gate, Kuroshio Publishing, 2010. (in Japanese)
- [7] Toshiyuki Sueyoshi, DEA-discriminant analysis in the view of goal programming, European Journal of Operational Research, 115, 564-582, 1999.

Design of Omni-directional Hexapod Robot with Horizontal Coxa Joint

1st Marek Zak

Brno University of Technology
Faculty of Information Technology
Centre of Excellence IT4Innovations
Brno 612 66, Czech Republic
Email: izakmarek@fit.vutbr.cz

2nd Jaroslav Rozman

Brno University of Technology
Faculty of Information Technology
Centre of Excellence IT4Innovations
Brno 612 66, Czech Republic
Email: rozmanj@fit.vutbr.cz

3rd Frantisek V. Zboril

Brno University of Technology
Faculty of Information Technology
Centre of Excellence IT4Innovations
Brno 612 66, Czech Republic
Email: zboril@fit.vutbr.cz

Abstract—Legged robots have great potential to travel across rough terrain, but they mostly move quite slow. This paper deals with the design and construction of a omni-directional seven degrees of freedom hexapod (i.e. six-legged) robot which is equipped with a horizontal coxa joint. The wheels can be used in flat terrain so the robot can reach higher speeds and additional coxa joint makes the robot more robust when climbing in rough terrain. The robot is controlled by terrain adaptive movement controller, which reacts on terrain difficulty and adjusts robot movement speed and gait.

Index Terms—Hexapod Robot; Hexapod Control; Movement Controller; Horizontal Coxa Joint

I. INTRODUCTION

Legged chassis, and especially hexapods, are one of the most robust robot chassis. They can operate in rough terrain, cross large holes and continue moving even after losing a leg or two [1]. However, their control is much more difficult and they are rather slower than wheeled chassis.

In this paper, we present a new combination of omni-directional and legged robot which can reach a higher speed than ordinary legged robots. Additionally, the robot is equipped with horizontal coxa joint, which provides better body support in inclined terrains.

The robot is controlled by terrain adaptive movement controller, which reacts on terrain difficulty and adjusts robot movement speed and gait.

During research in the field of walking robots many robots differing in shape, size, leg count or the number of joints per leg were built. Here are just few examples of the most interesting walking robots.

Takubo et al. introduced a four-degree-of-freedom hexapod robot Asterisk, which has the hexagonal body type and is capable of gripping things using two of its legs. It can also walk on a grid ceiling. Each limb switches between two functions: leg function for locomotion and arm function for manipulation [2].

Yoshioka et al. introduced Asterisk H, which is based on previous version Asterisk, but it is equipped with a wheel on the end of each leg. The robot can use both legged and wheeled locomotion [3].

ATHLETE is a six-legged walking robot designed by NASA. Its primary purpose is an exploration of planets

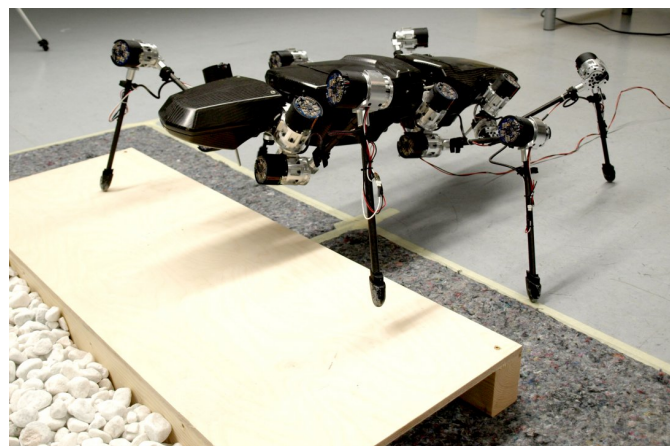


Fig. 1. Hector is six-legged robot and its construction was inspired by stick insect. It has passive elastic joints and an ultralight exoskeleton. Taken from [6].

especially of Mars [4]. It has hexagonal body type and its legs are evenly distributed across the body. Each leg is ended by a wheel. Thanks to this feature the robot can act as omni-directional chassis. Also, the robot can drive on easy terrain and walk once it runs into rough terrain. The robot can also attach a drill or gripper to its legs and use them as a manipulator.

Hector is a legged robot which combines compliant joint drives, a rich sensorization and decentralized control approaches to develop bio-equivalent walking capabilities [5]. Its construction was inspired by stick insect. It has passive elastic joints and an ultralight exoskeleton. The robot is in Figure 1.

LAURON V is fifth generation of a six-legged robot which is inspired by the stick insect. Thanks to new leg design it can cope with difficult obstacles and steep inclines and can be deployed for search and rescue missions [7].

The multi-legged autonomous eXplorer (MAX) is an ultralight 2.25 m tall hexapod robot. Its design emphasizes a low mass/size ratio and high locomotion efficiency [8].

Weaver is a five-degree-of-freedom hexapod robot which has proprioceptive control and exteroceptive terrain perception

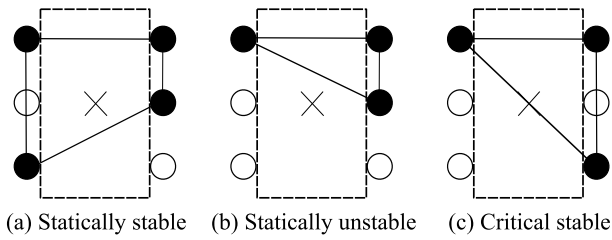


Fig. 2. Possible positions of the chassis during its movement. Black dots represent legs on the ground and symbol "x" represents the center of gravity. a) Robot is statically stable, because the center of gravity is inside the leg area. b) Robot is statically unstable, because the center of gravity is outside the leg area. c) Robot is critically stable, because the center of gravity is on the border of the leg area. This figure is taken from [10].

capabilities. It is equipped with two cameras which can detect terrain so the robot can adapt its locomotion parameters [9].

II. LEGGED ROBOTS

Legged robots are in the center of interest for a long time. Many walking robots which differ in body shape [11], number of legs, number of degrees of freedom per leg or locomotion technique were built. Various options can be combined to achieve many different configurations.

Walking chassis movements can be divided into statically stable and dynamically stable [1], [10]. Statically stable chassis (e.g. hexapod) can remain in a stable position in every moment of its movement. Dynamically stable chassis is sometimes out of balance – balancing or falling (e.g. bipedal).

During their movement, statically stable chassis are in one of the three basic states based on the number of legs and used gait (Figure 2) [10]. The first state is statically stable when the chassis rests on at least three legs and is in balance. This is usual for the chassis with more legs (e.g., hexapod) which is characterized by statically stable walking (the position of the chassis is stable in each moment). It can also be in statically unstable state when the chassis is not balanced which leads to collapse. This instability can be compensated by a dynamic movement. Then we talk about dynamically stable walking which is a typical example of bipedal chassis.

The last state, critically stable state, is between the previous two. In this state, chassis balances on the edge of its center of gravity. These features should be also considered when designing a control system.

A. Gaits

A gait refers to the locomotion achieved through the movement of robot legs. Compared to humans, the legged chassis usually has more than two legs. Therefore, the locomotion of a robot is much more complicated. There are several basic gaits, such as tripod, wave or ripple [12] (Figure 3).

The selection of gait has a great impact on the hexapod movement. While tripod gait is the fastest statically stable gait (the maximum number of legs - three - is lifted at once) [13], [14] and can be used mostly in flat terrains, wave gait is the

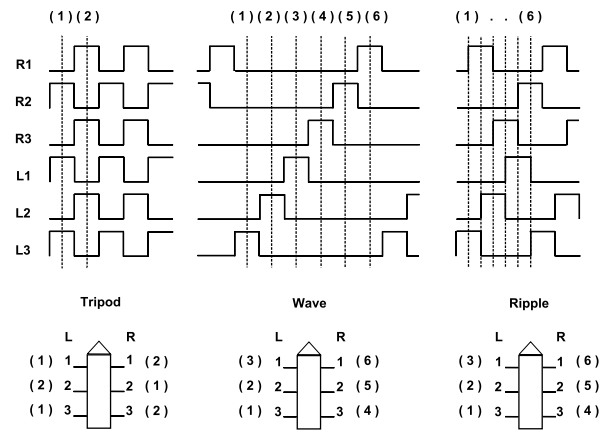


Fig. 3. Walking gaits. The chart shows the movement of each leg in time. A high value represents leg movement, low values means no movement. Tripod, wave and ripple gaits are shown in this figure. Tripod has two group of legs, all the legs in the same group move at once. In the wave gait only one leg is moving forward at any time. After all legs are set up to their new positions, step is completed. In the ripple gait all legs move the same way, but their moves are shifted. Inspired by <http://www.oricomtech.com/projects/cynthia2.gif>, 30. 9. 2015.

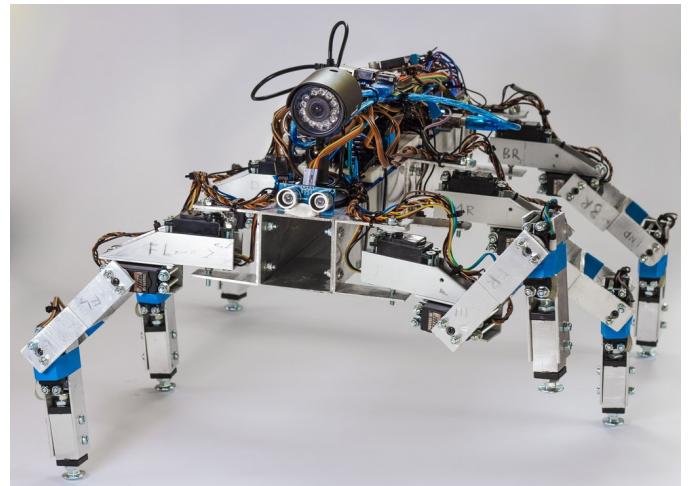


Fig. 4. The predecessor of our new robot is made of aluminium profiles. It is equipped with 18 hobby servomotors with encoders, inertial measurement unit, force-sensitive resistors to detect ground, ultrasonic sonars and camera to detect obstacles and environment. It is controlled by Arduino Mega board [15] and Raspberry Pi mini computer [16].

most statically stable gait (only one leg moves at a time) and provides the most robust movement in rough terrain. These two gaits can be combined in many ways so the gait keeps the higher speed but become more stable in difficult terrain conditions. Tetrapod is such a gait. It is modified wave gait, but it moves two legs instead of one at a time.

III. ROBOT DESIGN

The robot was designed as rectangular hexapod with seven degrees of freedom (DOF) per leg (five joints are used for positioning the leg and two joints are used as omni-directional chassis). Its design is based on our previous hexapod (see

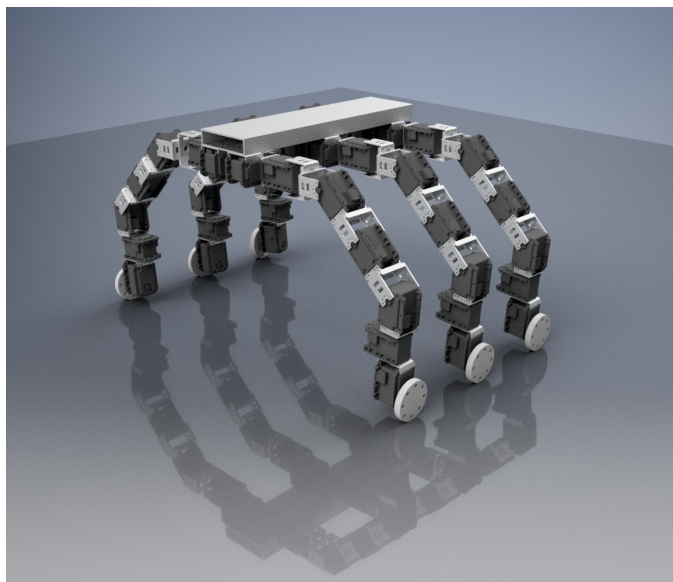


Fig. 5. 3D model of the robot. Each leg consists of seven servomotors. Coxa joint moves the leg forward and backward and femur, tibia and tarsus joints lift the leg up and down. Two servomotors are part of the omni-directional chassis and one servomotor can rotate the leg in a horizontal axes.

Figure 4) which has only three DOF, is made of aluminium profiles and uses common hobby servomotors. The 3D model of the new robot is shown in Figure 5.

Thanks to symmetric design the robot can continue moving even when it falls on its back. It can use either horizontal coxa joint or omni-directional rotate joint to set up its legs so it can keep moving. Also, the front and the back of the robot is interchangeable unless the robot is equipped with some special device (e.g. camera) on only one side.

The robot consists of 42 MX series Dynamixel servomotors [17] which are chained together using an asynchronous serial line. Each servomotor has full rotation mode and build-in encoders and thermometer. The robot is controlled by an Arduino Due board [18] which is built on the ARM Cortex-M3 chip [19]. The robot also has an inertial measurement unit (IMU) and is powered by two Li-Po 11.1 V accumulator. The length of the stretched leg is 45 cm, the maximum light height is 39 cm and the length of the robot is 58 cm.

A. Leg design

The design of robot leg determines what walking patterns can robot use. An ordinary hexapod needs at least two degrees of freedom (DOF) to be able to move forward. One to lift the leg up and down (coxa joint) and one to move the leg forward and backward (femur joint). But three degrees of freedom are recommended because using just two DOF will lead to slipping of the feet on the ground. The third joint (tibia joint) will ensure that the leg can move along the right trajectory [10], [20]. The basic configuration of three DOF leg is shown in Figure 6.

Although three DOF are enough, adding more DOF to the leg can help the robot manage more terrain difficulties.

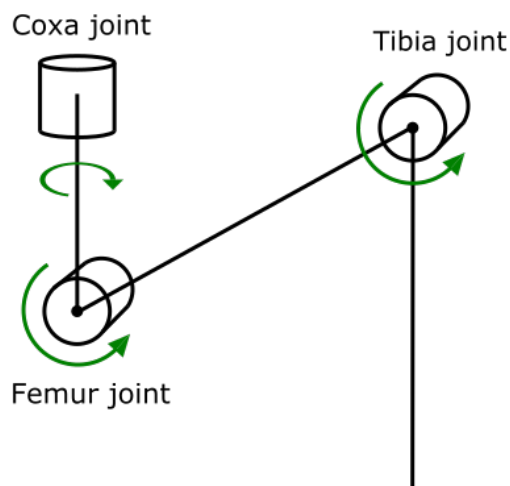


Fig. 6. Basic configuration of hexapods leg. From the left, coxa joint which moves the leg forward and backward, femur and tibia joints which move the leg up and down.

Therefore, our robot is equipped with five DOF. Except for those three basic joints the robot has tarsus joint and one horizontal coxa joint. Both these joints allow the robot climb more difficult terrains and can be used for object manipulation since five joints create simple robotic arm without a gripper. Moreover, the horizontal coxa joint stabilizes the robot when climbing sloping terrains (see Figure 7). Common legged robots must place their legs behind the body when climbing uphill. Our robot can rotate its legs using the horizontal coxa joint which leads to higher stability and lower energy consumption because the leg is parallel to the gravity vector. The leg is shown in Figure 8.

Each leg is equipped with an IMU, so the leg can be placed parallel to the gravity vector. It can also be used to detect anomalies in the leg motion or the orientation of the leg tip.

B. Omni-directional leg

One of the biggest disadvantages of small walking robots is their low speed. They can move only several meters per minute. On the other hand, their movement is really robust. To eliminate this disadvantage our robot is equipped with wheels on the end of each leg tip. This allows the robot to use wheels in flat terrains and switch to legs once it runs into rough terrain. Additionally, each wheel can rotate around the z-axis, so it creates omni-directional chassis.

C. Sensors

The robot is equipped with ultrasonic range meters which can detect obstacles around the robot. It also has an IMU on the body and on the foot tip of each leg so the robot could place legs parallel to the gravity vector and detect movement anomalies. Each servomotor has encoders, thermometer and voltage and torque indicators which can be used to measure

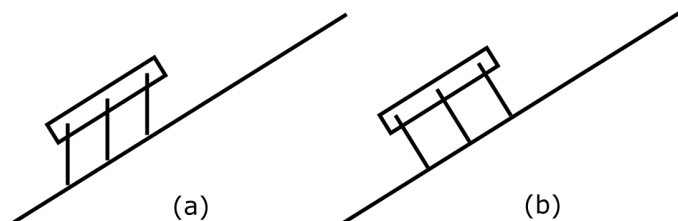


Fig. 7. Difference between hexapod with (a) and without (b) horizontal coxa joint. Hexapod with horizontal coxa joint can rotate its legs so the foot tip is parallel to the gravity vector. This reduces power consumption, better supports the body and allows the robot to climb more difficult and inclined terrains. This extra joint can also be used to turn the legs when the robot falls on its back.

the load on each leg. The robot can detect ground using force-sensitive resistors which are placed in each leg.

IV. ROBOT CONTROLLER

The controller consists of five main blocks: reflexive layer, fuzzy controller, gait selector, leg coordinator and leg controllers (see Figure 9). Additionally, it uses data from the sensor layer.

Reflexive layer uses sensors data to activate reflexes by sending direct commands into leg coordinator or leg controller. If no reflex is triggered, the sensors data goes into a fuzzy controller which transforms the data into terrain difficulty. This difficulty is then sent into gait selector which selects the most appropriate gait. Selected gait is then executed by leg coordinator which commands leg controllers. Each leg has one leg controller which is responsible for transforming leg coordinator commands into the desired movement of the leg.

A. Reflexive layer

The reflexive layer of the movement controller implements three basic reflexes.

The stepping reflex ensures, that the robot keeps the legs in the best positions to spare energy or to better support the body. If it is possible, the leg is moved closer to the body.

The elevator reflex is used when the leg is moving to a new position. If the leg encounters an obstacle and cannot finish its move, it tries to lift the leg higher and step over the obstacle.

Searching reflex is used when the leg cannot reach the ground at the expected location. It then tries to find another foothold to support the body and finish the step. Reflexes are shown in Figure 10.

Reflexive layer is also responsible for avoiding dynamic obstacles which are not part of high-level path planning, relocating the leg when it loses foothold and rotating the legs in inclined terrain or when the robot falls on its back.

B. Fuzzy controller

The fuzzy controller is used to transform data from the sensor layer into terrain difficulty which is then used by gait selector to pick up the most appropriate gait for current terrain conditions. The fuzzy controller uses data from IMU, encoders, torque meters, thermometers and ultrasonic sonars.

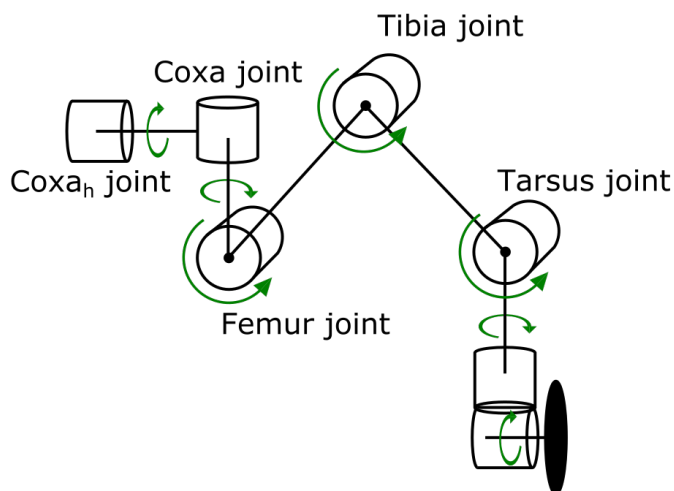


Fig. 8. Robot leg configuration. Coxa_h joint is used for leg rotation in y axis, so the robot legs can stay parallel to the gravity vector even when climbing sloping terrains. Coxa joint is used to move the leg forward and backward and femur, tibia and tarsus joints are used to move the leg up and down. Last two servomotors are used as omni-direction chassis, so the robot can reach higher speed in flat terrain conditions.

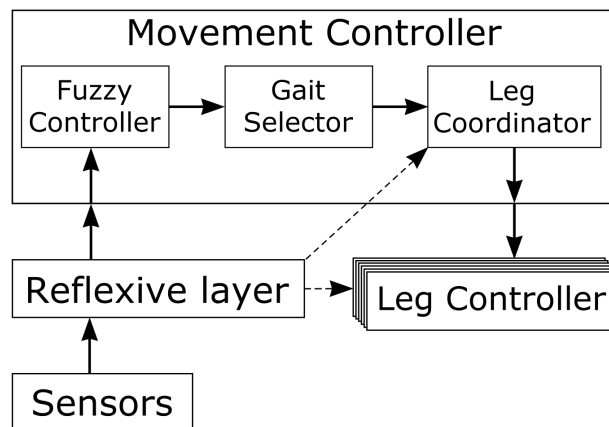


Fig. 9. A schema of robot controller. Sensors provides data to reflexive layer which can control leg movement directly in case of emergency. Sensor data is also sent as an input to fuzzy controller where are transformed and used by gait selector to determinate the most appropriate gait for current terrain. Selected gait is then executed by leg coordinator which commands leg controllers.

C. Gait selector

The gait selector uses information from the fuzzy controller and picks up the most suitable gait for current terrain conditions. Once the gait is selected the leg coordinator executes this gait.

D. Leg coordinator

The most suitable gait selected by the gait selector is then executed by the leg coordinator. Its task is to synchronize leg movements and compose selected gait from each leg movements. The leg coordinator can also be interrupted by reflexive layer.

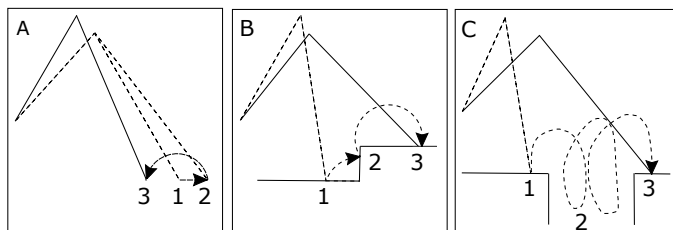


Fig. 10. A) Stepping reflex. The leg can step from the position 2 to the position 3 to better support the body. B) Elevator reflex. If the leg encounters an obstacle 2, it tries to lift the leg higher to step over the obstacle and find new position for the leg tip. C) Searching behaviour. If the leg cannot reach ground at the expected location 2, it tries to find another foothold 3. This figure is taken from [21].

E. Leg controller

Each leg has its leg controller which is managed by leg coordinator. The leg controller receives the instruction of speed and goal position from the leg coordinator. These values are converted to coordinates using inverse kinematic and the leg swing is made.

V. CONCLUSION

This paper dealt with the design of a hexapod robot which combines advantages of wheeled and legged chassis. Each leg has seven degrees of freedom - five for leg movement and two as an omni-directional chassis, which allows the robot to travel much faster than ordinary legged robots. Additionally, a horizontal coxa joint provides better body support in inclined terrains and allows the robot to operate after falling on its back. We also proposed a new design of movement controller, which can react on terrain difficulty and adjust robot movement speed and gait.

In the future work, we would like to add leg tips on the opposite side of tarsus joint to eliminate the possibility that the wheel gets stuck in the terrain inequality.

ACKNOWLEDGMENT

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602 and by the project IGA FIT-S-17-4014.

REFERENCES

- [1] U. Saranlı, "Dynamic locomotion with a hexapod robot," Ph.D. dissertation, The University of Michigan, 2002.
- [2] T. Takubo, T. Arai, K. Inoue, H. Ochi, T. Konishi, T. Tsurutani, Y. Hayashibara, and E. Koyanagi, "Integrated limb mechanism robot asterisk," *Journal of Robotics and Mechatronics*, vol. 18, no. 2, pp. 203–214, 2006.
- [3] T. Yoshioka, T. Takubo, T. Arai, and K. Inoue, "Hybrid locomotion of leg-wheel asterisk h," *Journal of Robotics and Mechatronics*, vol. 20, no. 3, pp. 403–412, 2008.
- [4] NASA, "All-terrain hex-limbed extra-terrestrial explorer," <http://athlete.jpl.nasa.gov/>, 2009, [Online; visited 30-09-2015].
- [5] A. Schneider, J. Paskarbeits, M. Schilling, and J. Schmitz, "Hector, a bio-inspired and compliant hexapod robot," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2014, pp. 427–429.
- [6] CITEC, "First steps for hector the robot stick insect," <https://www.citec.de/en/news/first-steps-hector-robot-stick-insect>, 2019, [Online; visited 13-09-2019].
- [7] A. Rönnau, G. Heppner, M. Nowicki, and R. Dillmann, "Lauron v: A versatile six-legged walking robot with advanced maneuverability," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2014, pp. 82–87.
- [8] A. Elfes, R. Steindl, F. Talbot, F. Kendoul, P. Sikka, T. Lowe, N. Kottege, M. Bjelonic, R. Dungavell, T. Bandyopadhyay *et al.*, "The multilegged autonomous explorer (max)," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1050–1057.
- [9] M. Bjelonic, N. Kottege, T. Homberger, P. Borges, P. Beckerle, and M. Chli, "Weaver: Hexapod robot for autonomous navigation on unstructured terrain," *Journal of Field Robotics*, vol. 35, no. 7, pp. 1063–1079, 2018.
- [10] S. Manoiu-Olaru, M. Nitulescu, and V. Stoian, "Hexapod robot. mathematical support for modeling and control," in *System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on*, Oct 2011, pp. 1–6.
- [11] E. Moore and M. Buehler, "Stable stair climbing in a simple hexapod robot," DTIC Document, Tech. Rep., 2001.
- [12] F. Tedeschi and G. Carbone, "Design issues for hexapod walking robots," *Robotics*, vol. 3, no. 2, pp. 181–206, 2014.
- [13] D. M. Wilson, "Insect walking," *Annual review of entomology*, vol. 11, no. 1, pp. 103–122, 1966.
- [14] D. Belter and P. Skrzypczyński, "A biologically inspired approach to feasible gait learning for a hexapod robot," *International Journal of Applied Mathematics and Computer Science*, vol. 20, no. 1, pp. 69–84, 2010.
- [15] Arduino, "Arduino mega 2560 rev3," <https://store.arduino.cc/mega-2560-r3>, 2019, [Online; visited 10-09-2019].
- [16] R. P. M. B., "Raspberry pi," <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>, 2019, [Online; visited 30-08-2019].
- [17] ROBOTIS, "Dynamixel - mx series," <http://www.robotis.us/mx-series/>, 2019, [Online; visited 02-09-2019].
- [18] Arduino, "Arduino due," <https://store.arduino.cc/due>, 2019, [Online; visited 02-09-2019].
- [19] Atmel, "Atsam3x8e," http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel1105732bitCortexM3MicrocontrollerSAM3X-SAM3A_Datasheet.pdf, 2019, [Online; visited 03-09-2019].
- [20] X. Ding, A. Rovetta, J. Zhu, and Z. Wang, *Locomotion analysis of hexapod robot*. INTECH Open Access Publisher, 2010.
- [21] K. S. Espenschied, R. D. Quinn, R. D. Beer, and H. J. Chiel, "Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot," *Robotics and autonomous systems*, vol. 18, no. 1, pp. 59–64, 1996.

AUTHOR INDEX

- | | | | |
|------------------------|----------|----------------------|----------|
| Abe, Y. | 467 | Hrkút, P. | 429 |
| Albert, A. | 271 | Hudák, M. | 160, 375 |
| Al-sabti, K. | 338 | Huraj, L. | 165 |
| Andrešič, D. | 13 | Huszaník, T. | 171, 266 |
| Avdagic, Z. | 43 | Chodarev, S. | 175 |
| Bachratá, K. | 181 | Chovanec, M. | 181 |
| Bachratý, H. | 181 | Ioannou, A. | 309, 315 |
| Bajer, D. | 18 | Iovanovici, A. | 188 |
| Bajtoš, T. | 260 | Jakab, F. | 198 |
| Bako, M. | 456 | Janech, J. | 193 |
| Baláz, A. | 70 | Jasenčáková, K. | 181 |
| Ballagi, Á. | 155 | Kaihara, T. | 461, 467 |
| Bažant, M. | 294 | Kainz, O. | 198 |
| Bednár, B. | 26 | Karpiel, E. | 198 |
| Békefi, B. F. | 31 | Kekeňák, T. | 321 |
| Berta, P. | 37 | Kello, T. | 100 |
| Besic, I. | 43 | Kholeif, M. | 343 |
| Bilanová, Z. | 26, 70 | Kiss, A. | 87, 216 |
| Bogdándy, B. | 49 | Kohút, R. | 228 |
| Brunner, T. | 54, 299 | Kokuryo, D. | 461, 467 |
| Bučková, H. | 60 | Kordíć, S. | 450 |
| Cervenka, M. | 357, 445 | Korečko, Š. | 160 |
| Cosic, J. | 66 | Kossecki, P. | 204 |
| Čerešník, P. | 70 | Kovács, A. | 31 |
| Čibej, U. | 76, 277 | Kozsik, T. | 110 |
| Čibej, U. | 277 | Krammer, P. | 210 |
| Dimitrieski, V. | 450 | Kršák, E. | 100, 106 |
| Dobravec, T. | 82 | Kruppai, G. | 216 |
| Donkó, I. | 87 | Kurkowski, M. | 1, 369 |
| Dostál, J. | 60, 387 | Kvassay, M. | 210 |
| Dudáš, A. | 93 | Kvet, M. | 435 |
| Dudjak, M. | 18 | Kvet, M. | 193, 222 |
| Dudova, J. | 100 | Lang, J. | 228 |
| Ďuračík, M. | 106, 429 | Łazarska, M. | 234 |
| Duží, M. | 271 | Leba, M. | 188 |
| Eged, J. | 380 | Lucká, M. | 351 |
| Erdenebat, B. | 110 | Lukács, D. | 242 |
| Fathalla, R. | 343 | Luković, I. | 450 |
| Fujii, N. | 461, 467 | Madeja, M. | 248 |
| Fülöp, E. | 116 | Madoš, B. | 70 |
| Gaceanu, R. D. | 399 | Maimaris, A. | 309, 315 |
| Galíneck, D. | 121 | Majcher, F. | 375 |
| Gawronska, E. | 128 | Marcek, D. | 254 |
| Gombos, G. | 87 | Marek, J. | 294, 333 |
| Goubey, M. | 405 | Marková, E. | 260 |
| Grzybowski, A. Z. | 134, 139 | Martinović, G. | 18 |
| Gulabovska, H. | 144 | Márton, M. | 266 |
| Habala, O. | 210 | Matiaško, K. | 222 |
| Halama, J. | 175 | Mazur, M. | 1 |
| Haluška, R. | 150 | Melicherčík, M. | 327, 441 |
| Haratim, M. | 387 | Menšík, M. | 271 |
| Hluchý, L. | 210 | Meško, M. | 106, 429 |
| Hodzic, K. | 43 | Mézešová, T. | 260, 321 |
| Hollósi, J. | 155 | Mihelič, J. | 76, 277 |
| Horváthová, D. | 456 | | |

Michalíková, A.	283	Skala, V.	357, 363, 445
Michalko, M.	198	Smolik, M.	363, 445
Mitter, M.	456	Soboň, A.	369
Morog, D.	66	Sobota, B.	160, 375
Moscovici, A.	188	Sokol, P.	260, 321
Myslivec, T.	405	Stachowiak, S.	369
Nečasová, G.	288	Starczewski, T.	139
Nedvědová, M.	294, 333	Steingartner, W.	121, 380, 387
Németh, B.	299	Suchánová, B.	13
Ness, A. N.	309, 315	Sulír, M.	393
Novitzká, V.	380	Szabados, K.	31
Olenčin, M.	304	Szalai-Gindl, J. M.	87
Ovseník, L.	150, 171, 266	Szederjesi-Dragomir, A.	399
Pajr, M.	271	Šaloun, P.	13
Papageorgiou, G.	309, 315	Šátek, V.	288
Pataki, N.	116	Šimon, M.	165
Patschka, V.	271	Škrinárová, J.	93
Pažický, B.	283	Šul'aj, P.	150
Pekarčík, P.	321	Švejda, M.	405
Pellegrini, L.	188	Tamás, J.	411
Perhác, J.	304	Tavač, M.	193
Petija, R.	198	Tejfel, M.	242
Pisarić, M.	450	Teplická, K.	417
Pongrácz, G.	242	Tikhonenko, O.	423
Pop, H. F.	399	Toth, Š.	429
Porkoláb, Z.	54, 144	Tóth, Z.	49
Porubän, J.	248, 393	Turán, J.	171, 266
Povinský, M.	327, 441	Urbanský, J.	266
Pozdílková, A.	294, 333	Václavková, M.	435
Puchała, P.	134	Vagač, M.	441
Radaković, D.	380	Vasta, J.	445
Ristić, S.	450	Veigend, P.	288
Rozman, J.	473	Vesel, E.	93
Ružbarský, J.	106	Vještica, M.	450
Sagi, G.	338	Voštinár, P.	456
Salem, I. A.	343	Vranić, V.	37
Santo, R.	467	Wang, X.	461
Sarbu, C.	399	Watanabe, R.	467
Sedláček, P.	435	Zaymus, E.	429
Schlehuber, C.	66	Zbořil, F. V.	473
Schreiner, W.	7	Zebić, V.	121
Siedlecka-Lamch, O.	234	Ziółkowski, M.	423
Siládi, V.	327	Zorić, B.	18
Sitarčík, J.	351	Žák, M.	473
Sivý, M.	375		