Zbornik 25. mednarodne multikonference

INFORMACIJSKA DRUŽBA

Zvezek A

Proceedings of the 25th International Multiconference

INFORMATION SOCIETY

Volume A

2022

Slovenska konferenca o umetni inteligenci

Slovenian Conference on Artificial Intelligence

Uredniki • Editors:
Mitja Luštrek, Matjaž Gams, Rok Piltaver

Ljubljana, Slovenija
11. oktober

11 October
Ljubljana, Slovenia

http://is.ijs.si

Zbornik 25. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2022

Zvezek A

Proceedings of the 25th International Multiconference

INFORMATION SOCIETY – IS 2022

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredniki / Editors

Mitja Luštrek, Matjaž Gams, Rok Piltaver

http://is.ijs.si

11. oktober 2022 / 11 October 2022
Ljubljana, Slovenija

Uredniki:

Mitja Luštrek
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Matjaž Gams
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Rok Piltaver
Outfit7
in Odsek za inteligentne sisteme, Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2022

Informacijska družba
ISSN 2630-371X

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID 127433219
ISBN 978-961-264-241-9 (PDF)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2022

Petindvajseta multikonferenca *Informacijska družba* je preživela probleme zaradi korone. Zahvala za skoraj normalno delovanje konference gre predvsem tistim predsednikom konferenc, ki so kljub prvi pandemiji modernega sveta pogumno obdržali visok strokovni nivo.

Pandemija v letih 2020 do danes skoraj v ničemer ni omejila neverjetne rasti IKTja, informacijske družbe, umetne inteligence in znanosti nasploh, ampak nasprotno – rast znanja, računalništva in umetne inteligence se nadaljuje z že kar običajno nesluteno hitrostjo. Po drugi strani se nadaljuje razpadanje družbenih vrednot ter tragična vojna v Ukrajini, ki lahko pljuskne v Evropo. Se pa zavedanje večine ljudi, da je potrebno podpreti stroko, krepi. Konec koncev je v 2022 v veljavo stopil not raziskovalni zakon, ki bo izboljšal razmere, predvsem leto za letom povečeval sredstva za znanost.

Letos smo v multikonferenco povezali enajst odličnih neodvisnih konferenc, med njimi »Legende računalništva«, s katero postavljamo nov mehanizem promocije informacijske družbe. IS 2022 zajema okoli 200 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic ter 400 obiskovalcev. Prireditev so spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica (http://www.informatica.si/), ki se ponaša s 46-letno tradicijo odlične znanstvene revije. Multikonferenco Informacijska družba 2022 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Izkopavanje znanja in podatkovna skladišča
- Demografske in družinske analize
- Kognitivna znanost
- Kognitonika
- Legende računalništva
- Vseprisotne zdravstvene storitve in pametni senzorji
- Mednarodna konferenca o prenosu tehnologij
- Vzgoja in izobraževanje v informacijski družbi
- Študentska konferenca o računalniškem raziskovanju
- Matcos 2022

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi ACM Slovenija, SLAIS, DKZ in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in institucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

S podelitvijo nagrad, še posebej z nagrado Michie-Turing, se avtonomna stroka s področja opredeli do najbolj izstopajočih dosežkov. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe je prejel prof. dr. Jadran Lenarčič. Priznanje za dosežek leta pripada ekipi NIJZ za portal zVEM. »Informacijsko limono« za najmanj primerno informacijsko potezo je prejela cenzura na socialnih omrežjih, »informacijsko jagodo« kot najboljšo potezo pa nova elektronska osebna izkaznica. Čestitke nagrajencem!

Mojca Ciglarič, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2022

The 25th *Information Society Multiconference* (http://is.ijs.si) survived the COVID-19 problems. The multiconference survived due to the conference chairs who bravely decided to continue with their conferences despite the first pandemics in the modern era.

The COVID-19 pandemic from 2020 till now did not decrease the growth of ICT, information society, artificial intelligence and science overall, quite on the contrary – the progress of computers, knowledge and artificial intelligence continued with the fascinating growth rate. However, the downfall of societal norms and progress seems to slowly but surely continue along with the tragical war in Ukraine. On the other hand, the awareness of the majority, that science and development are the only perspective for prosperous future, substantially grows. In 2020, a new law regulating Slovenian research was accepted promoting increase of funding year by year.

The Multiconference is running parallel sessions with 200 presentations of scientific papers at eleven conferences, many round tables, workshops and award ceremonies, and 400 attendees. Among the conferences, “Legends of computing” introduce the “Hall of fame” concept for computer science and informatics. Selected papers will be published in the Informatica journal with its 46-years tradition of excellent research publishing.

The Information Society 2022 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Data Mining and Data Warehouses
- Cognitive Science
- Demographic and family analyses
- Cognitonics
- Legends of computing
- Pervasive health and smart sensing
- International technology transfer conference
- Education in information society
- Student computer science research conference 2022
- Matcos 2022

The multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, SLAIS, DKZ and the second national academy, the Slovenian Engineering Academy. In the name of the conference organizers, we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

The award for life-long outstanding contributions is presented in memory of Donald Michie and Alan Turing. The Michie-Turing award was given to Prof. Dr. Jadran Lenarčič for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, the yearly recognition for current achievements was awarded to NIJZ for the zVEM platform. The information lemon goes to the censorship on social networks. The information strawberry as the best information service last year went to the electronic identity card. Congratulations!

Mojca Ciglarič, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia
Sergio Campos-Cordobes, Spain
Shabnam Farahmand, Finland
Sergio Crovella, Italy

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Blaž Mahnič

Programme Committee

Mojca Ciglarič, chair
Bojan Orel,
Franc Solina,
Viljan Mahnič,
Cene Bavec,
Tomaž Kalin,
Jozsef Györkös,
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Mitja Luštrek
Marko Grobelnik

Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček

Andrej Ule
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah
Niko Zimic
Rok Piltaver
Toma Strle
Tine Kolenik
Franci Pivec
Uroš Rajkovič
Borut Batagelj
Tomaž Ogrin
Aleš Ude
Bojan Blažica
Matjaž Kljun
Robert Blatnik
Erik Dovgan
Špela Stres
Anton Gradišek

KAZALO / TABLE OF CONTENTS

Zbornik 25. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2022

Zvezek A

Proceedings of the 25th International Multiconference

INFORMATION SOCIETY – IS 2022

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredniki / Editors

Mitja Luštrek, Matjaž Gams, Rok Piltaver

http://is.ijs.si

11. oktober 2022 / 11 October 2022
Ljubljana, Slovenija

PREDGOVOR

Umetna inteligenca še vedno hitro napreduje, so pa glavni dosežki lanskega leta na področjih, kjer smo jih že vajeni. Avtonomna vozila so vedno bolj avtonomna in se že uporabljajo za prevoz potnikov, čeravno v zelo omejenem obsegu. Jezikovni modeli, kot so izboljšani GPT-3, postajajo zreli za praktično uporabo, zato se njihovi stvaritelji začenjajo ukvarjati s tem, kako jih odvračati od tvorbe politično nekorektnih besedil. Po eni strani razumljivo, po drugi strani pa – ob problemu omejevanja svobode govora na spletu, ki si je letos prislužil nominacijo za informacijsko limono – tudi nekoliko skrb zbujajoče. Modeli za generiranje slik iz opisov, katerih prvi vidnejši predstavnik je bil DALL-E, so se letos namnožili, in videli smo več poizkusov njihove uporabe za izdelavo stripov. Potlej pa so tu še aplikacije v robotiki, medicini, računalniški varnosti in seveda zvitemu streženju spletnih reklam.

Ko umetna inteligenca postaja vedno zmožnejša in bolj razširjena, se pojavljalo pomisleki o njeni varnosti ter prizadevanja za uporabo, ki bo družbi v korist in ne v škodo. Ta škoda se začne z nepotrebnimi nakupi zaradi preveč zvitih reklam, ki nas spremljajo že dolgo in smo se z njimi sprijaznili, nadaljuje pa s še resnejšimi problemi, kot so denimo slabe medicinske in zaposlovalne odločitve. Zaradi tovrstnih problemov vse več držav sprejema zakonodajo o umetni inteligenci, ki bo raziskovalcem bržkone povzročila nekaj sivih las, a če bo dobra – in k temu skušajmo prispevati, kolikor lahko – bo tudi pomagala, da naše delo ne bo dobilo zloveščega pridiha. Vse več je tudi razmišljanja o splošni umetni inteligenci z zmožnostmi, ki presegajo človeške. Njen vpliv na človeško družbo utegne biti dramatičen. A če želimo zagotoviti, da bo dramatično dober, se bomo morali v prihodnjih letih resno lotiti raziskovalnega področja zagotavljanja, da kompleksni modeli umetne inteligence zares počno tisto, kar mislimo in želimo, da počno, ki je zaenkrat še precej v povojih.

Za konec pa poglejmo, kako je letos z našo konferenco. 11 prispevkov, ki smo jih prejeli, sicer ne opisuje tako visokoletečega dela, kot ga obravnavata prejšnja dva odstavka, so pa vseeno kakovosti in morda začetek česa pomembnega. Število je zmerno in Institut Jožef Stefan še malo bolj prevladujoč, kot običajno, za kar do neke mere krivimo COVID-19 – ne ker bi nas še vedno hudo pestil, ampak ker sta dve konferenčno klavrni leti raziskovalce konferenčenja malo odvadili. A upajmo, da bo tudi to minilo. Prirejamo pa letos v okviru konference Data Science Meetup – dogodek z lepo tradicijo in dobro udeležbo, kjer imajo strokovnjaki iz industrije kratke predstavitve svojega dela. Na to smo ponosni, saj rešuje težavo pomanjkanja prispevkov iz industrije, ki smo se je dotaknili že v preteklih predgovorih.

FOREWORD

Artificial intelligence is still making good progress, but the major achievements of the past year are in the areas where we have grown to expect them. Autonomous vehicles are increasingly autonomous and already being used to carry passengers, albeit in a very limited way. Language models, such as the improved GPT-3, are becoming ready for practical use. Because of that, their authors are starting to work on preventing them from generating politically incorrect texts. This is on one hand understandable, but on the other hand – considering the problem of censorship on the internet, which was nominated for the Information Lemon this year – somewhat concerning. Models that generate images from text descriptions, whose first prominent representative was DALL-E, are proliferating. We have seen several attempts of using them to generate comics. There are also applications in robotics, medicine, cybersecurity and of course cunning delivery of online ads.

With artificial intelligence becoming ever more capable and pervasive, concerns about its safety and use for the benefit of the society rather them harm are increasingly raised. The harm starts with unnecessary consumption due to insidious advertising, but these is old news we have become accustomed to. However, there are potentially more serious problems, such as bad medical or employment decisions. Because of these, a number of countries are drafting legislation about artificial intelligence. This will surely be a headache for researchers, but if the legislation is good – and we should help make it such if we can – it will benefit the reputation of our work. Superhuman general artificial intelligence is also increasingly entering professional and public debate. Its impact on the humanity could be dramatic. To ensure it is dramatically good, we will have to tackle the very much open research problem of ensuring that complex artificial-intelligence models indeed do what we think and want them to do.

Let us finally take a look at our conference. The 11 papers we received are not describing work as ambitious as that described in the previous paragraphs, but they are nevertheless good and perhaps the beginning of something important. The number is modest and Jožef Stefan Institute even more overrepresented than usual, which we partially blame on COVID-19. Not that it is still a major problem, but in the two years without truly good conference the researchers seem to have lost the habit of going to conferences to some degree. We hope that this, too, shall pass. On a brighter note, we are organizing Data Science Meetup as a part of our conference. This is an event with a longstanding tradition and good attendance in which experts from the industry give short talks on their work. We are quite proud of this achievement, since it addresses the lack of papers from the industry which we bemoaned in past forewords.

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Mitja Luštrek

Matjaž Gams

Rok Piltaver

Cene Bavec

Marko Bohanec

Marko Bonač

Ivan Bratko

Bojan Cestnik

Aleš Dobnikar

Erik Dovgan

Bogdan Filipič

Borka Jerman Blažič

Marjan Krisper

Marjan Mernik

Biljana Mileva Boshkoska

Vladislav Rajkovič

Niko Schlamberger

Tomaž Seljak

Peter Stanovnik

Damjan Strnad

Miha Štajdohar

Vasja Vehovar

Martin Žnidaršič

Initial Results in Predicting High-Level Features of Constrained Multi-Objective Optimization Problems

Andrejaana Andova
Aljoša Vodopija
Jožef Stefan Institute and
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
andrejaana.andova@ijs.si
aljosa.vodopija@ijs.si

Pavel Krömer
Vojtěch Uher
Department of Computer Science
VSB - Technical University of
Ostrava
17. listopadu 2172/15
Ostrava-Poruba, Czech Republic
pavel.kromer@vsb.cz
vojtech.uher@vsb.cz

Tea Tušar
Bogdan Filipič
Jožef Stefan Institute and
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
tea.tusar@ijs.si
bogdan.filipic@ijs.si

ABSTRACT

Trying numerous algorithms on an optimization problem that we encounter for the first time in order to find the best-performing algorithm is time-consuming and impractical. To narrow down the number of algorithm choices, high-level features describing important problem characteristics can be related with algorithm performance. However, characterizing optimization problems for this purpose is challenging, especially when they include multiple objectives and constraints. In this work, we use machine learning (ML) to automatically predict high-level features of constrained multi-objective optimization problems (CMOPs) from low-level, exploratory landscape analysis features. The results obtained on the MW benchmark show a significant difference in classification accuracy depending on the applied evaluation approach. The poor performance of the leave-one-problem-out strategy indicates the need for further investigation of the relevance of low-level features in CMOP characterization.

KEYWORDS

constrained multi-objective optimization, exploratory landscape analysis, sampling methods, problem characterization, machine learning

1 INTRODUCTION

Predicting high-level features of constrained multi-objective optimization problems (CMOPs) is important as it can help decide which algorithm to use when faced with a new (real-world) CMOP. The structure of the objective and constraint functions are usually unknown for such problems. Moreover, the evaluation of problem solutions might be very time-consuming. In such cases, it is beneficial to know certain high-level features of the CMOP, which eases the selection of an appropriate multi-objective optimization algorithm or constraint handling technique to solve the problem efficiently.

Two frequently considered high-level features of CMOPs are the problem type and connectivity of the feasible region. The problem type characterizes whether and how the constraints change the Pareto front of the problem. As pointed out by Tanabe et al. [8], this feature is useful as it indicates whether the problem needs to be treated as constrained or unconstrained. Moreover, Ma et al. [5] showed which constraint handling techniques are more successful in solving CMOPs, depending on the problem type. Similarly, the connectivity of the feasible region (or problem connectivity for short) defines the multimodality of the problem violation landscape and, therefore, crucially affects the choice of algorithms that can solve the problem efficiently [5].

High-level features of a new problem can be predicted using automatically calculated low-level problem features. The most widely known low-level features in evolutionary optimization are the exploratory landscape analysis (ELA) features. They were initially introduced to characterize single-objective optimization problems and implemented in the flacco package [2]. More recently, Liefooghe et al. [4] proposed a set of ELA features for multi-objective optimization problems, and Vodopija et al. [10] introduced additional ELA features for CMOPs.

In this work, we use the ELA features from [4] and some from [10] to investigate whether they are useful for predicting problem type and connectivity. To the best of our knowledge, this is the first attempt to predict the high-level features of CMOPs. A similar study was performed by Renau et al. [7] on single-objective optimization problems. They used ELA features to classify the optimization problem. When splitting the data into training and test sets, instances from the same problem were used for both training and testing. The first of our three experiments follows this setup. However, because this evaluation methodology is not useful in practice (the class of a new real-world problem is unknown), a second experiment is performed using the leave-one-problem-out methodology. Finally, the third experiment varies the number of target problem instances used for training to gain further insight in the difficult task of predicting high-level features from low-level ones.

The paper is further organized as follows. In Section 2, we introduce the theoretical background of constrained multi-objective optimization. In Section 3, we explain the features used in this study. In Section 4, we present the considered test problems, and in Section 5 the experimental setup. In Section 6, we report on the obtained results. Finally, in Section 7, we provide a conclusion and present the ideas for future work.

2 THEORETICAL BACKGROUND

A CMOP can be formulated as:

$$\begin{aligned} \text{minimize} \quad & f_m(x), \quad m = 1, \ldots, M \\ \text{subject to} \quad & g_k(x) \le 0, \quad k = 1, \ldots, K, \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_D)$ is a *search vector* of dimension D, $f_m : S \rightarrow \mathbb{R}$ are *objective functions*, $g_k : S \rightarrow \mathbb{R}$ *constraint functions*,

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

$S \subseteq \mathbb{R}^D$ is the *search space*, and M and K are the numbers of objectives and constraints, respectively.

A solution x is *feasible*, if it satisfies all constraints $g_k(x) \leq 0$ for $k = 1, \ldots, K$. For each constraint g_k we can define the *constraint violation* as $v_k(x) = \max(0, g_k(x))$. The *overall constraint violation* is defined as

$$v(x) = \sum_{i}^{K} v_k(x). \tag{2}$$

A solution x is feasible iff $v(x) = 0$.

A feasible solution $x \in S$ is said to *dominate* another feasible solution $y \in S$ if $f_m(x) \leq f_m(y)$ for all $1 \leq m \leq M$, and $f_m(x) < f_m(y)$ for at least one $1 \leq m \leq M$. A feasible solution $x^* \in S$ is a *Pareto-optimal solution* if there exists no feasible solution $x \in S$ that dominates x^*. All feasible solutions constitute the *feasible region*, $F = \{x \in S \mid v(x) = 0\}$, and all nondominated feasible solutions form the *Pareto set*, S_o. The image of the Pareto set in the objective space is the *Pareto front*, $P_o = \{f(x) \mid x \in S_o\}$.

3 EXPLORATORY LANDSCAPE ANALYSIS

ELA is a selection of techniques able to analyze the search and objective space of a problem, their correlation and their characteristics with the goal of identifying the features important for the performance of optimization algorithms. To extract the ELA features, one needs to first generate a sample of solutions. The ELA features use statistical methods to characterize the problem landscape. Thus, one can use an arbitrary sample size. However, the ELA features are generally more accurate for large sample sizes. The ELA features proposed by Liefooghe et al. [4] and used also in this work can be divided into four categories: global, multimodality, evolvability, and ruggedness features.

The global features capture certain global problem properties, for example, the correlation between the objective values, average and maximum distance between solutions in the search space and the objective space, the proportion of non-dominated solutions, the average and maximum rank of solutions, etc.

The multimodality features assess the number of local optima in the objective space. They are computed for the bi-objective space and also for each objective separately, in both cases by analyzing the neighbourhood of each solution. If a solution dominates its neighbors (or has a better objective value than its neighbors), it is defined as a local optimum. The multimodality features comprise the proportion of solutions that are locally optimal, the average and maximum distances between local optima, etc.

The evolvability features describe how fast a local optimizer would converge towards an optimum. They are calculated by analyzing how many neighboring solutions are dominated by, dominating, or incomparable with a given solution.

The ruggedness features measure the correlation between the information and quality from neighboring solutions – larger correlation means a smoother landscape. The features are calculated by using Spearman's correlation coefficient on the evolvability features between each pair of neighboring solutions.

In addition, we include four ELA features from [10] that describe the violation landscape and its relation with the objective space. The first feature is the feasibility ratio. It is expressed as the proportion of feasible solutions in the sample and is one of the most frequently used features in categorizing violation landscapes. The second feature is the maximum value of overall constraint violation values in the sample. The last two features measure the relationship between the objectives and constraints. They are the minimum and maximum correlations between the objectives and the overall constraint violation.

Table 1: High-level features of the MW test problems.

Problem	Type	Connectivity
MW1	II	Disconnected
MW2	I	Disconnected
MW3	III	Connected
MW4	I	Connected
MW5	II	Connected
MW6	II	Disconnected
MW7	III	Connected
MW8	II	Disconnected
MW9	IV	Connected
MW10	III	Disconnected
MW11	IV	Disconnected
MW12	IV	Disconnected
MW13	III	Disconnected
MW14	I	Connected

4 TEST PROBLEMS

We base this study on 14 CMOPs proposed by Ma et al. [5] and called MW1–14. In addition to proposing the problems, the authors also describe them with high-level features, such as the problem type and connectivity of the feasible region. The values of these two high-level features for each MW problem are listed in Table 1.

Many of the ELA features proposed by Liefooghe et al. [4] can only be calculated for bi-objective optimization problems. Therefore, we investigate only the bi-objective versions of the MW problems although three of them are scalable in the number of objectives. All MW problems are also scalable in the number of variables. We use 5-dimensional problems to match the experimental setup from [7].

5 EXPERIMENTAL SETUP

In preliminary experiments, we used six sampling methods from the ghalton [1] and scipy [9] Python libraries: gHalton, Halton, Sobol, Latin hypercube sampling, optimized Latin hypercube sampling, and uniform sampling [3]. The results have shown that similar prediction accuracies are obtained when using data provided by any of these sampling methods. For this reason, we only present the results obtained using the Sobol sampling method in the rest of the paper.

The Sobol sampling method generates a sample set by partitioning the search space and filling each partition with a sample solution. We generate additional Sobol sample sets using the Cranley-Patterson rotation [3]. The solutions from the original sample set are rotated using a random shift of each dimension, thus creating new sample sets that preserve the properties of the Sobol sampling. The modulo operation keeps the shifted values within the unitary interval. This approach was also used by Renau et al. [7].

Following this approach, we generate 100 sets of samples, each with 512 solutions, which we then evaluate on all 14 MW benchmark problems. For each problem and sample set pair, we compute 46 ELA features, which represent a single instance in the data. As a result, by evaluating the 100 sample sets on each of the 14 test problems, we get 1400 data instances. We then use these data instances and the corresponding high-level problem features (problem type and connectivity) to train a classifier for predicting the high-level problem features.

We use two widely used machine learning (ML) methods for classification: the Random Forest (RF) classifier and the k-Nearest Neighbors (KNN) classifier. The reason for choosing these classifiers instead of some others is that, usually, RF performs favorably compared to other ML classifiers. KNN, on the other hand, uses the distance between solutions as a performance metric, which is useful when analyzing the obtained classification results visually. For both RF and KNN, we apply the implementation from the scikit-learn library [6]. For KNN, we keep the default settings, while for RF we train 100 trees.

We perform three experiments that differ in the classifier evaluation methodology. In the first experiment, we base the evaluation methodology on the work by Renau et al. [7], where the data is split by using instances from the same problem for both training and testing. There, 50% of all instances are used for training, and the remaining 50% for testing. Furthermore, we take care of dividing the instances into training and test sets so that the proportion of instances from each problem is equal in both sets.

However, this methodology does not correspond to the real-world scenario where we want to learn the high-level features of a problem encountered for the first time. Therefore, we use the leave-one-problem-out evaluation methodology in the second experiment. Here, the instances from a single problem are used for testing, and the instances from all other problems for training. The procedure is repeated for all problems and the classification accuracy is calculated as the average over all train-test splits.

Finally, the third experiment is performed to see how adding target problem data to the training set influences the resulting classification accuracy. In this experiment, we vary the percentage of target problem data that is used for training between 0% and 99% with the step of 1%. When it equals 0%, no target problem data is used for training, which corresponds to the leave-one-problem-out methodology of the second experiment. Note that this setup never equals the one from the first experiment because here the data of all other (non-target) problems is always used for training. Again, this procedure is repeated for all problems and we report the average classification accuracy.

To better understand the task we are trying to solve, we visualize the classes by first reducing the dimensionality of the feature space from 46-D to 2-D using Pairwise Controlled Manifold Approximation Projection (PaCMAP) [11]. We use the Python package pacmap with default parameter values.

6 RESULTS

The results of the first experiment, where 50% of all data is used for training and 50% for testing, show that both RF and KNN achieve a classification accuracy above 98% (see Table 2). An explanation for such good results can be derived from the two leftmost plots in Figure 1. Here, we can see that PacMAP finds many clusters in the data. However, the clusters are highly correlated to the problems themselves. Thus, leaving some instances from the target problem in the training set results in a high classification accuracy because the classification task is now transformed into identifying to which cluster the new sample belongs, which is a much easier task to perform.

Table 2: Classification accuracy when 50% of all data is used for training and 50% for testing (first experiment).

Learning method	Problem type	Problem connectivity
RF	98%	99%
KNN	100%	100%

The more realistic scenario of having to predict the high-level feature of a yet unseen problem is tested in the second experiment. Here, the classification accuracy drops to only 7–19% for the problem type prediction, and to 41–57% for the problem connectivity prediction (see the leftmost points corresponding to 0% on the plots in Figure 2). This is comparable to the classification accuracy of the stratified classifier, which achieves 19% for the problem type prediction and 45% for the problem connectivity prediction. We can look at the results of the third experiment to help us understand this decline in classification accuracy. As seen from Figure 2, adding just a few instances of the target problem to the training set drastically increases the classification accuracy.

When the training data contains no instances from the target problem, the classifier is forced to find information about the high-level feature from other problems. However, this is a much harder task given that similar problems often have different high-level features (see the middle and right plots in Figure 1).

In the visualizations in Figure 1 the points indicating the correctly classified instances have black edges. As we can see, for many problems, RF has a 0% classification accuracy (top middle and top right plot). There are, however, some problems for which RF finds the correct class for a number of instances. Nonetheless, from these 2-D plots it is hard to understand why certain instances are misclassified by RF. This is because RF detects details in the data that the dimensionality reduction visualization method is unable to capture.

Similar behavior can be observed for KNN. Given that KNN classifies an instance depending on the classes of its most similar instances, the visualization from Figure 1 can help interpret its poor results on the leave-one-problem-out methodology. We can see that the clusters created by PacMAP are not well-aligned with the high-level features of problem type and connectivity. This makes predicting them a hard task for KNN. The clustering by PacMAP suggests that the applied ELA features are not descriptive enough for predicting problem type and connectivity.

7 CONCLUSION AND FUTURE WORK

In this work, we tried to predict high-level features of CMOPs. More specifically, using low-level ELA features, we constructed the classifiers to predict the problem type and connectivity. Two ML classifiers were utilized, RF and KNN.

We employed three evaluation methodologies. The first one follows the related work and splits the data into two halves, one serving as the training set and the other as the test set (instances from the same problem are used in both sets). The second evaluation methodology uses all instances from the target problem for testing, and none for training. The third method gradually adds the target problem data to the training set. We achieved excellent classification accuracy with the first evaluation methodology, but very poor ones with the second one. The drop in classification accuracy was checked by the third methodology, which has shown that already a small number of instances of the same problem increases the classification accuracy.

Visualizations of the data in the form of 2-D plots show that CMOP instances form clusters that are highly correlated to the problem instances, but not to the high-level problem features. For

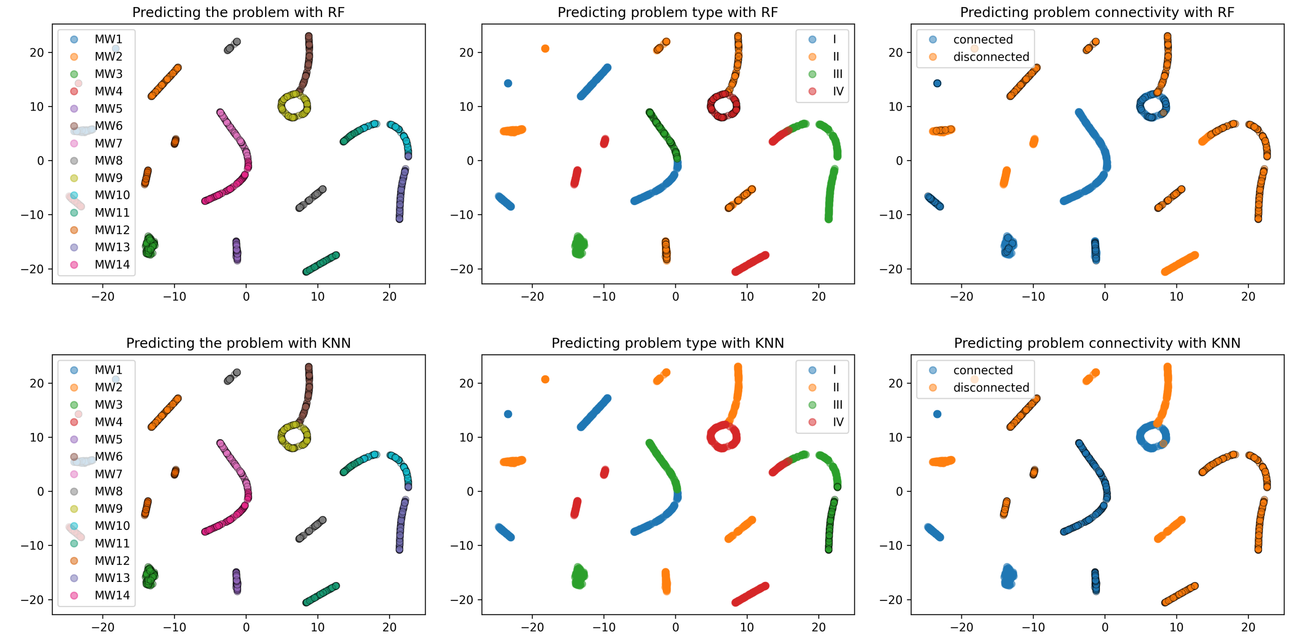


Figure 1: Dimensionality reduction of the ELA feature space using the PacMAP method. Points are colored based on their true values with correct classifications denoted by a black point edge. The top and bottom rows show the results for Random Forest and KNN, respectively, while the different classification targets are arranged in columns: the left column displays the results for the problem, the middle for problem type and the right for problem connectivity.

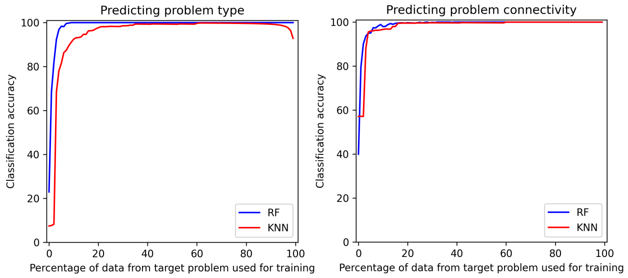


Figure 2: Classification accuracy for different proportions of data from the target problem used for training.

this reason, by including some instances from the target problem in the training set, the classification task becomes an easier task of recognizing to which cluster an instance belongs. Unfortunately, this is not a realistic scenario, since in the real world we have no information on the characteristics of the newly encountered problem. We therefore recommend to use the second evaluation methodology when addressing this task.

However, the initial results obtained using the second evaluation methodology are not so promising. A possible improvement could be considering more ELA features in the learning procedure, either additional ones from [10] or newly created ones. Moreover, using a more representative set of test problems from various benchmark suites may also improve classifier performance.

ACKNOWLEDGMENTS

The authors acknowledge the project "Constrained multi-objective Optimization Based on Problem Landscape Analysis" was financially supported by the Slovenian Research Agency (project no. N2-0254) and the Czech Science Foundation (grant no. GF22-34873K). The Slovenian authors acknowledge additional financial support from the Slovenian Research Agency (young researcher program and research core funding no. P2-0209).

REFERENCES

[1] François-Michel De Rainville, Christian Gagné, Olivier Teytaud, and Denis Laurendeau. 2012. Evolutionary optimization of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, 22, 2, 1–25.

[2] Christian Hanster and Pascal Kerschke. 2017. flaccogui: Exploratory landscape analysis for everyone. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Companion*. ACM, 1215–1222.

[3] Christiane Lemieux. 2009. *Monte Carlo and Quasi-Monte Carlo Sampling*. *Springer Series in Statistics*. Springer New York, NY.

[4] Arnaud Liefooghe, Sébastien Verel, Benjamin Lacroix, Alexandru-Ciprian Zăvoianu, and John McCall. 2021. Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 421–429.

[5] Zhongwei Ma and Yong Wang. 2019. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23, 6, 972–986.

[6] F. Pedregosa et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[7] Quentin Renau, Carola Doerr, Johann Dreo, and Benjamin Doerr. 2020. Exploratory landscape analysis is strongly sensitive to the sampling strategy. In *International Conference on Parallel Problem Solving from Nature*. Springer, 139–153.

[8] Ryoji Tanabe and Akira Oyama. 2017. A note on constrained multi-objective optimization benchmark problems. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1127–1134.

[9] Pauli Virtanen et al. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272.

[10] Aljoša Vodopija, Tea Tušar, and Bogdan Filipič. 2022. Characterization of constrained continuous multiobjective optimization problems: A feature space perspective. *Information Sciences*, 607, 244–262.

[11] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. 2021. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22, 201, 1–73.

Learning the Probabilities in Probabilistic Context-Free Grammars for Arithmetical Expressions from Equation Corpora

Marija Chaushevska
marija.chaushevska@ijs.si
Jožef Stefan Int. Postgraduate School
& Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Ljupčo Todorovski
ljupco.todorovski@fmf.uni-lj.si
Jožef Stefan Institute &
Faculty of Mathematics and Physics
Jadranska cesta 21
Ljubljana, Slovenia

Jure Brence & Sašo Džeroski
jure.brence@ijs.si|saso.dzeroski@ijs.si
Jožef Stefan Institute &
Jožef Stefan Int. Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

A core challenge for both physics and artificial intelligence (AI) is symbolic regression: finding a symbolic expression that matches data from an unknown function. Symbolic regression approaches are largely data-driven and search an unconstrained space of mathematical expressions, often employing genetic algorithms. On the other hand, equation discovery approaches incorporate domain knowledge to constrain the structure space and search it using local or exhaustive search methods. In this paper, we adopt the use of probabilistic context-free grammars (PCFG) in equation discovery and propose a method for learning the probabilities of production rules in such PCFGs. We take a universal PCFG with an initial set of manually assigned probabilities for each production rule. We learn new probabilities by parsing each expressions in a given corpus of expression, such as the Feynman dataset.

KEYWORDS

equation discovery, grammar, probabilistic context-free grammar, parsing, learning probabilities, probability distribution

1 INTRODUCTION

Equation discovery is an area of machine learning that develops methods for automated discovery of quantitative laws, expressed in the form of equations, in collections of measured numeric data [5] [11]. More precisely, equation discovery methods seek to automate the identification of equation structure as well as parameters. Traditionally, domain experts derive equation structure based on the theory in the domain and use standard numerical optimization methods to estimate their parameters. Equation discovery methods often use domain knowledge to specify the space of equations they consider. The key questions in the field are how to best represent the symbolic language of mathematics, how to incorporate domain knowledge in the process of equation discovery, as well as how to perform the search for optimal equation structures. Symbolic regression methods are largely data-driven and search an unconstrained space of mathematical expressions, often employing evolutionary algorithms. On the other hand, equation discovery methods, such as process-based modeling [4], incorporate domain knowledge to constrain the structure space and search using greedy-local [12] or exhaustive search methods on the constrained space. The task of equation discovery is closely related to the task of supervised regression. Machine learning methods for supervised regression assume a fixed class of models, such as linear regression or neural networks with a particular architecture, and find the one that provides the best fit to the training data. Equation discovery methods typically consider broader classes of mathematical equations. These classes may be vast and many (often infinitely many) equations can be found that provide excellent fit to the training data. The challenge of symbolic regression is therefore twofold. On one hand, one can easily overfit the training data with an unnecessarily complex equation. On the other hand, the space of candidate equations is huge and grows exponentially as equation complexity increases, posing serious computational issues to equation discovery methods.

Equation Discovery systems explore the hypothesis space of all equations that can be constructed given a set of arithmetic operators, functions and variable. They search for equations that fit given input data best. The number of all possible candidate equations can be infinite.

Early equation discovery systems used parametric approaches to specify the space of polynomial equations considered. LAGRAMGE [13] uses context-free grammars (CFG) [9] to specify the language of equations considered. The recent system ProGED [2] uses probabilistic context-free grammars (PCFG), where a probability is associated with each production rule. In this paper, we propose a method for learning these probabilities for a given PCFG by using a given corpus of expressions.

2 GRAMMARS FOR EQUATION DISCOVERY

A grammar is a finite specification of a language. A language can contain an infinite number of strings, or even if it is finite, it can contain so many strings that it is not practical to list them all. Originating from computational linguistics, grammars are used as formal specifications of languages and use a set of production rules to derive valid strings in the language of interest. A grammar mainly consists of a set of production rules, rewriting rules for transforming strings. Each rule specifies a replacement of a particular string (its left-hand side) with another (its right-hand side). A rule can be applied to each string (equation) that contains its left-hand side and produces a string in which an occurrence of that left-hand side has been replaced with its right-hand side. A grammar further distinguishes between two kinds of symbols: non-terminal and terminal symbols; each left-hand side must contain at least one non-terminal symbol. It also distinguishes a special non-terminal symbol, called the start symbol. In equation discovery, we are interested in using grammars as generative models, as opposed to their common use for parsing, i.e., discriminating between legal and illegal strings in a language.

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

2.1 Context-Free Grammar (CFG)

In formal language theory, a context-free grammar [9] is a formal grammar which is defined as a tuple G = (N, T, R, S). It is used to generate all possible patterns of strings in a given formal language. The syntax of the expression on the right-hand side of the equation is prescribed with a context-free grammar. The set T contains terminal symbols, i.e.,words for composing sentences in the language or variables in the arithmetical expressions. The terminals are primitive grammar symbols that can not be further rewritten, i.e., no productions are affiliated with them. Non-terminal symbols (syntactic categories) in the set N represent higher-order terms in the language, such as noun or verb phrases. Each of the non-terminals represents expressions or phrases in a language. The production rules in the set R are rewrite rules of the form A $\rightarrow \alpha$, where the left-hand side is a non-terminal, $A \in N$, while the right-hand side is a string of non-terminals and terminals, $\alpha \in (N \cup T)^*$. In natural language, a rule $NP \rightarrow AN$ specifies that a noun phrase NP has an adjective A and a noun N. A and N represent the subsets of adjectives and nouns, which are both terminals. No matter which symbols surround it, the single non-terminal on the left hand side can always be replaced by the right hand side. This is what distinguishes it from context-sensitive grammar. When deriving a sentence, a grammar starts with a string containing a single non-terminal $S \in N$ and recursively applies production rules to replace non-terminals in the current string with the strings on the right-hands sides of the rules. The final string contains only terminal symbols and belongs to the language defined by G.

In equation discovery, grammars represent sets of expressions that can appear in the right hand side of equations. These grammars use several symbols with special meanings. For example, the terminal *const* $\in T$ is used to denote a constant parameter in an equation that has to be fitted to the input data.

A simple context-free grammar $G_M = (N_M, T_M, R_M, S_M)$ deriving linear expressions from variables x, y, z is as follows:

$$\begin{aligned} N_M &= \{E, V\} \\ T_M &= \{x, y, z, +, *\} \\ R_M &= \{E \rightarrow E + V | E * V | V \\ V &\rightarrow x|y|z\} \\ S_M &= E \end{aligned} \tag{1}$$

We write multiple production rules with the same non-terminal on the left hand side using a compact, single-line representation, e.g., $E \rightarrow E + V \mid E * V \mid V$ stands for the set of rules $\{E \rightarrow E + V, E \rightarrow E * V, E \rightarrow V\}$.

2.2 Probabilistic Context-Free Grammar (PCFG)

Grammar formalisms are not new to the field of equation discovery [4] [3] [13], but probabilistic grammars are. A probabilistic grammar assigns probabilities to productions and thereby allows one to use the grammar as a stochastic generator [10] [6] [15]. Probabilistic Context-Free Grammars (PCFGs), are a simple model of phrase-structure trees. They extend context-free grammars (CFGs) similarly to how hidden Markov models extend regular grammars. A grammar can be turned into a probabilistic grammar by assigning probabilities to each of its productions, such that for each $A \in N$:

$$\sum_{(A \rightarrow \alpha) \in R} P \rightarrow (A \rightarrow \alpha) = 1 \tag{2}$$

The probability of a derivation (parse) is defined as the product of the probabilities of all the production rules used to expand each node in the parse tree (derivation). These probabilities can be viewed as parameters of the model. The probabilities of all productions with the same non-terminal symbol on the left hand side sum up to one, i.e., we impose a probability distribution on the productions with the same symbol on the left hand side. As each parse tree, derived by a grammar G, is characterized by a sequence of productions, its probability is simply the product of the probabilities of all productions in the sequence [11].

We can extend the example context-free grammar G_M above to a PCFG by assigning a probability to each of the six productions, given below in brackets after each production:

$$\begin{aligned} E &\rightarrow E + V[p] | E * V[q] | V[1 - p - q] \\ V &\rightarrow x[p_v] | y[q_v] | z[1 - p_v - p_q] \end{aligned} \tag{3}$$

Here we have parameterized the probability distributions over productions for E and V with the parameters $0 < p < 1; 0 < q < 1; 0 < p_v < 1$; and $0 < q_v < 1$, respectively.

Context-free grammars are typically used to parse sentences. Probabilistic context-free grammars provide an estimate of the probability of a parse tree, in addition to the tree itself. Probabilistic context-free grammars also allow for another type of application — stochastic generation of sentences or, in our case, mathematical expressions. The probabilities, assigned to the productions, provide a great amount of control over the probability distribution of individual parse trees. In our example in Eq. 3, the parameters p and q control the probability of a larger number of terms in an expression, while the parameters p_v and q_v tune the ratio between the number of occurrences of variables x, y and z.

An important concept to consider when working with grammars is ambiguity. A grammar is formally ambiguous if there exist sentences (expressions) that can be described by more than one parse tree, generated by the grammar. Grammars for arithmetic expressions can express another type of ambiguity, called semantic ambiguity. All but the simplest arithmetic expressions can be written in many mathematically equivalent, but grammatically distinct ways. It is generally useful to adopt a canonical representation that each generated equation is converted into. This allows us to compare expressions to each other and check whether they are mathematically equivalent in addition to comparing their parse trees. In our work, we use the Python symbolic mathematics library SymPy [8] to simplify expressions, convert them into canonical form, and compare them symbolically.

3 LEARNING PROBABILITIES IN PCFGS FOR ARITHMETICAL EXPRESSIONS

Parameter learning approaches for PCFGs assume a fixed set of production rules and try to learn the probabilities assigned to them. Some approaches encourage sparsity and remove rules with very small probabilities. Parameter learning approaches are typically more scalable than structure search approaches, because parameter learning is a continuous optimization problem which is in general easier than the discrete optimization problem of structure search. Therefore, most of the state-of-the art algorithms for unsupervised learning of natural language grammars are parameter learning approaches.

3.1 The Approach

In this paper, we propose a parameter learning approach for PCFGs, based on parsing a corpus of expressions. We adopt the universal PCFG probabilistic context-free grammar for arithmetic expressions used by ProGED [2]. While ProGED uses manually assigned probabilities in this grammar, we use an initial set of randomly assigned probabilities to each production rule. The universal grammar is composed of production rules that include the four basic operations (+,-,*, /), basic functions (such as sin or log), constant parameters and variables.

Our method for learning probabilities from a given corpus of expressions is designed on the assumption that the probability of a production rule in the grammar is proportional to the incidence of the production in the parse trees for the expressions in the corpus. It uses a parser from the NLTK (Natural Language Toolkit) Python library [1] to parse the expressions in the given corpus using the universal PCFG. NLTK contains classes to work with PCFGs and there are different types of parsers implemented in the NLTK Python library. In particular, we use the InsideChartParser(), a bottom-up parser for PCFG grammars that tries edges in descending order of the inside probabilities of their trees. The "inside probability" of a tree is simply the probability of the entire tree, ignoring its context. In particular, the inside probability of a tree generated by production p with children $c[1], c[2], ..., c[n]$ is $P(p)P(c[1])P(c[2])...P(c[n])$; and the inside probability of a token is 1 if it is present in the text, and 0 if it is absent. For a given string (expression) and a grammar, the parser determines whether the string can be derived using the grammar and if yes, returns the appropriate parse tree. After parsing the equations we count the number of times each production rule appears in the set of parsing trees, for all parsed equations (except for rules directly resulting in terminal symbols (variables)). We then group production rules by left non-terminal symbol and derive the probabilities for each production rule as the number of appearances of a given production rule divided by the sum of such numbers for all production rules for the same non-terminal.

3.2 The Corpora

We apply the proposed approach to two corpora of expressions (that appear on the right hand side of equations). The first one is the Feynman Symbolic Regression Database, which includes a diverse sample of equations from the three-volume set of physics textbooks by Richard P. Feynman [7] and has been previously used as a benchmark for equation discovery [14]. It was constructed by Udrescu and Tegmark [3] to facilitate the development and testing of algorithms for symbolic regression. The equations from Feynman database contain between one and nine variables, the four basic operations ($+, -, /, *$), the functions exp, $\sqrt{}$, sin, cos, tanh, arcsin and ln, as well as a variety of constants – mostly rational, but also e and π. There are three components to an arithmetic expression: variables, constants and operators. Numerical values and constants are typically treated as free parameters (terminal symbols) to be optimized when evaluating an equation for its fit against given data. We replaced all constants, such as e, π and rational constants with the terminal 'C'(const), because we treat them as free parameters. The minimum number of constants ('C'), in the Feynman database is 0, which means that there are a some equations that have only variables as terminal symbols. On the other hand the maximum number of constants in the Feynman database is five constants in only one equation.

The second corpus consists of 4080 scientific expressions from Wikipedia. Those mathematical expressions are named after people and they are parsed from Wikipedia. Compared to the Feynman dataset, Wikipedia's corpus contains more functions such as: Abs, $factorial$, tan, $sinh$, $cosh$ and pow (which do not exist in the Feynman database) as well as irrational constants (e and π) and numerical constants, which have to be replaced by a constant 'C'(const) in the grammar. The equations in Wikipedia's dataset contain between one and fifteen variables, which is twice as much compared to the Feynman dataset and the maximum number of 'C' terminal symbols is 16 per equation.

3.3 The Learned Probabilities

By using the proposed approach on the two corpora of arithmetic expressions described above, we obtain two sets of probabilities, with each probability assigned to one of the production rules in the PCFG. More precisely, we now have three universal PCFGs: (1) with the initial probabilities, manually assigned by the authors of ProGED, (2) with probabilities fine-tuned (learned) on the Feynman dataset, and (3) with probabilities fine-tuned (learned) on the Wikipedia corpus of arithmetical expressions.

In this section, we first present the three sets of probabilities, for each of the above mentioned PCFs: these are given in Table 1. We then compare the probability distributions across the rules for each non-terminal symbol (S,F,T and R) in the PCFGs.

As compared to the initial grammar, the grammar learned on the Feynman database reduces the probabilities of the recursive production rules ($S \rightarrow S + F$ and $S \rightarrow S - F$) and increases the probability of the non-recursive rule ($S \rightarrow F$): This leads to simpler expressions with fewer additive terms. In contrast, the grammar learned on the Wikipedia corpus has a probability for the rule $S \rightarrow S + F$ very comparable with the probability in the initial grammar. It also decreases the probability of the recursive production rule $S \rightarrow S - F$ and increases the probability of the non-recursive rule $S \rightarrow F$ by approximately 0.1 in each case.

The probabilities of the recursive production rules for the F non-terminal symbol ($F \rightarrow F * T$ and $F \rightarrow F/T$) are mostly larger than the ones in the initial grammar. An exception is the rule $F \rightarrow F/T$ with the Wikipedia corpus. The probability of the non-recursive production rule ($F \rightarrow T$) is smaller, slightly for the Wikipedia corpus, more substantially for the Feynman dataset.

In the learned probability distributions over the production rules for the non-terminal T, the probability of the rule $T \rightarrow V$ is much higher (goes from 0.4 to 0.7). In both learned grammars, the probabilities of the $T \rightarrow R$ and $T \rightarrow$ 'C' production rules are substantially reduced. This is more noticeable for $T \rightarrow$ 'C', where the probability goes from 0.4 to slightly above 0.1.

We finally discuss the probability distributions over the production rules for the non-terminal symbol R in the initial grammar and the two learned grammars. A probability with value 0 for a production rule here means that that function for the particular production rule is not present either in the Feynman corpus or the Wikipedia corpus of mathematical expressions. For example, the functions ln and $arcsin$ are not present in the arithmetical expressions from the Wikipedia dataset, but are present in the Feynman database. On the other hand, the functions log, pow, Abs, $sinh$, $cosh$, $factorial$ and tan do not exist in the arithmetic expressions from the Feynman database, that's why their probability is 0. The grammar learned on the Feynman database increases the probabilities of the production rules $R \rightarrow (S)$, $R \rightarrow sin(S)$ and $R \rightarrow sqrt(S)$ as compared to the probabilities of the initial grammar. In contrast, the probabilities of the remaining production

Table 1: Probabilities of the production rules for the non-terminal symbols in the initial grammar, the grammar trained on the Feynman database and the grammar trained on the Wikipedia corpus of expressions.

Production rule	Initial	Feynman	Wikipedia
S -> S + F	0.2	0.1034	0.2004
S -> S - F	0.2	0.1552	0.1108
S -> F	0.6	0.7414	0.6888
F -> F * T	0.2	0.3635	0.3349
F -> F / T	0.2	0.2446	0.1098
F -> T	0.6	0.3919	0.5553
T -> R	0.2	0.1554	0.1746
T -> ‘C’	0.4	0.1338	0.1174
T -> V	0.4	0.7108	0.7082
R →(S)	0.3	0.5391	0.6841
R → sin(S)	0.1	0.113	0.0249
R → arcsin (S)	0.1	0.0173	0
R → ln (S)	0.1	0.0087	0
R → tanh (S)	0.1	0.0087	0.0045
R → cos (S)	0.1	0.0956	0.0435
R → sqrt (S)	0.1	0.1304	0.0831
R → exp (S)	0.1	0.0872	0.0780
R → log(S)	0	0	0.0479
R → Abs(S)	0	0	0.0211
R → (S) ‘**’ (S)	0	0	0.0032
R → sinh(S)	0	0	0.0032
R → cosh(S)	0	0	0.0026
R → factorial(S)	0	0	0.0019
R → tan(S)	0	0	0.0019

rules learned on the Feynman database have lower probabilities as compared to the initial grammar. The grammar learned on the Wikipedia corpus increases the probability of the $R \to (S)$ production rule and decreases the probabilities of the remaining rules as compared to the initial grammar.

4 CONCLUSIONS AND FURTHER WORK

In this paper, we have proposed an approach to learn the parameters, i.e., production rule probabilities, in probabilistic context-free grammars for arithmetic expressions. We demonstrated the proposed approach by learning the probabilities in a universal grammar for arithmetic expressions from two corpora of expressions. The learned probabilities differ substantially from their initial values. Most notably, the initial settings underestimated the frequency of variables in favor of numerical constants, overestimated the need for recursion with addition and subtraction, while setting the probability of recursion with multiplication too low. These observations show how difficult it is to set probabilities manually and highlight the utility of the learning algorithm.

The comparison of the learned probability values for the two corpora also hints towards differing properties of the two collections of equations. The Wikipedia corpus seems to favor multiplication over division to a greater extent than the Feynman dataset. In terms of expression complexity, we observed a preference for high-order terms in the Feynman dataset, in contrast to a preference for higher numbers of low-order terms in the Wikipedia corpus. The observed differences between the properties of the two corpora demonstrate that the universal grammar is expressive enough to encode these properties and that the learning algorithm is able to discover them. The results show a great deal of promise for the goals of inferring domain knowledge from equation corpora and improving the efficiency of grammar-based equation discovery through the fine-tuning of production probabilities.

As further work we would like to perform equation discovery experiments using the three universal grammars: the universal grammar with initial (default) probabilities, with probabilities learned on the Feynman dataset and probabilities learned on the Wikipedia corpus. For this purpose, we will use the equation discovery system ProGED, which uses a Monte-Carlo approach of sampling equation structures from a given PCFG and evaluating their fit to the given data. We expect that the number of successfully reconstructed equations from the Feynman dataset, when using the learned (fine-tuned) universal PCFGs, will be higher as the number of equations successfully reconstructed with the universal grammar with manually set probabilities.

REFERENCES

[1] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python.* (1st ed.). O’Reilly Media, Inc.

[2] Jure Brence, Ljupčo Todorovski, and Sašo Džeroski. 2021. Probabilistic grammars for equation discovery. *Knowledge-Based Systems*, 224, (Apr. 2021), 1–12.

[3] Will Bridewell and Pat Langley. 2010. Two kinds of knowledge in scientific discovery. *Topics in Cognitive Science*, 21, 36–52.

[4] Will Bridewell, Pat Langley, Ljupčo Todorovski, and Sašo Džeroski. 2008. Inductive process modeling. *Machine Learning*, 71, 1, 1–32.

[5] Sašo Džeroski, Pat Langley, and Ljupco Todorovski. 2007. *Computational Discovery of Scientific Knowledge.* Vol. 4660. (Aug. 2007), 1–14.

[6] Brian C. Falkenhainer and Ryszard S. Michalski. 1990. Integrating quantitative and qualitative discovery: the abacus system. In *Machine Learning.* Yves Kodratoff and Ryszard S. Michalski, editors. Morgan Kaufmann, San Francisco (CA), 153–190.

[7] R.P. Feynman, R.B. Leighton, and M. Sands. 2015. *The Feynman Lectures on Physics, Vol. I: The New Millennium Edition: Mainly Mechanics, Radiation, and Heat.* Basic Books.

[8] Aaron Meurer et al. 2017. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, (Jan. 2017), e103.

[9] Alan P. Parkes. 2008. *A Concise Introduction to Languages and Machines.* (1st ed.). Springer Publishing Company, Incorporated.

[10] Cullen Schaffer. 1993. Bivariate scientific function finding in a sampled, real-data testbed. In *Machine Learning*, 167–183.

[11] Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *Science*, 324, 5923, 81–85.

[12] Jovan Tanevski, Ljupčo Todorovski, and Sašo Džeroski. 2020. Combinatorial search for selecting the structure of models of dynamical systems with equation discovery. *Engineering Applications of Artificial Intelligence*, 89, (Mar. 2020), 103423.

[13] Ljupčo Todorovski and Sašo Dzeroski. 1997. Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning* (ICML ’97). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 376–384.

[14] Silviu-Marian Udrescu and Max Tegmark. 2020. Ai feynman: a physics-inspired method for symbolic regression. *Science Advances*, 6, 16, eaay2631.

[15] R. Zembowitz and J Zytkow. 1992. Discovery of equations: experimental evaluation of convergence. In *Proceedings of Tenth National Conference on Artificial Intelligence.* Morgan Kaufmann Publishers Inc., San Mateo, CA, USA, 101–117.

Prediction of the Inflow in Macedonian Hydropower Plants, Using Machine Learning

Emilija Kizhevska
emilija.kizhevska@ijs.si
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Hristijan Gjoreski
hristijang@feit.ukim.edu.mk
Ss. Cyril and Methodius University
Faculty of Electrical Engineering
and Information Technologies
Rugjer Boskovic 18
Skopje, R.N.Macedonia

Mitja Luštrek
mitja.lustrek@ijs.si
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

As weather conditions become more complex and unpredictable as a consequence of global warming and air pollution, humans find it increasingly difficult to predict the amount of precipitation in the coming period, thus predicting the inflow into hydroelectric basins. Different types of hydropower plants (HPP), soil composition, how dry the soil is at the moment, the composition of precipitation, etc., also influence the inflow, making it even more difficult to be predicted. This research looks into the problem of predicting inflow in hydroelectric basins in Republic of North Macedonia and building machine learning models to do so. The main contribution of this research is the models for the largest five hydropower plants in Republic of North Macedonia (RM) that could optimize the loss and shortage of purchased electricity. Historical data from the closest meteorological station to each hydropower plant that we were working on, as well as historical data from the inflows at the hydropower plants, were used to build regression models that predict the inflow one day in advance for each hydropower plant separately. After deriving 19 new features, of which the majority are statistical, the predictive models' error was reduced. In the final step, we analyze the results empirically and qualitatively and comparing the models generated using different machine learning algorithms. For instance, one of the best models is the model for HPP Vrben, with the mean absolute error of around 8% of the average daily inflow. We built models using eight different regression algorithms for each hydropower plant, including linear regression and gradient boosting regression models as the models that make the smallest errors in predicting. These models could also help to prevent river and lake overflows in areas where hydropower facilities are located, with timely warnings minimizing the severity of natural disasters.

KEYWORDS

machine learning, regression models, hydropower plants, optimization of hydroelectric energy loss

1 INTRODUCTION

The global trend for producing electricity from renewable sources is increasing at an exponential rate. On the other hand, the world aims to reduce world's electricity losses, as there are more and more electrical devices and less and less non-renewable electricity sources. One of the solutions is to optimize electricity losses due to erroneous power consumption forecasts. First, from the standpoint of world non-renewable electricity savings consumption, as the coal or oil are, and then from the standpoint of public spending, because the later you purchase electricity, the more expensive it becomes. [3][5].

Hydropower now provides around 6.5% of the world's electricity needs. In Republic of North Macedonia the total installed capacity of hydropower is 556.8 MW, which is over 40% of the total capacity, ranking first among renewable energy sources. Hydroelectricity is used the most to meet daily variations in electricity consumption and to provide system services for regulation, allowing the power system to be more flexible and reliable. The peaks of electricity consumption are always regulated (i.e. supplemented) by hydroelectric energy while coal-fired power plants produce the majority of electricity. Predicting the quantity of electricity available from each hydropower plant in the future (the longer the period, the better) leads not only to the most efficient use of finances, but also to protection from natural disasters such as river and lake overflows. The amount of inflow in the foreseeable time to be predicted by humans becomes increasingly difficult, if not impossible, as meteorological conditions become more complex and unpredictable as a result of global warming and air pollution [4][1][9].

The inflow prediction in hydropower plants is mainly based on human judgement, however, it is not always accurate. Primarily, because it is about nature. There are two major issues with predicting future inflow accurately:

(1) Weather forecasting inaccuracy in the coming days. The issue is that the forecasting models get less accurate as the forecast gets further out in time [6].
(2) Different types of hydropower plants, especially the construction of pumped-storage hydropower plants, the complexity of geology, etc. In this research, we consider run-of-river and storage hydropower plants. Run-of-river hydropower plant includes a facility that channels flowing water from a river through a canal or pen-stock to spin a turbine, and rain influences the inflow almost immediately. Storage hydropower plants that include a dam and a reservoir to accumulate water, which is stored and released later when needed, providing flexibility to generate electricity on demand and reducing dependence on the variability of inflow. But whatever the hydropower plant type, the inflow is not directly related to weather conditions. For instance, the snow and hail do not accumulate straight away; it does require a time of melting, soil wetting, and for storage plants, additionally, conducting the water through the pipes to reach the basin, etc. [7].

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

The inflow into hydroelectric basins can be predicted more easily when using machine learning than by analyzing geology, satellite monitoring, pollution monitoring, and changes in global warming. Developing a machine learning model that connects all of these features, allows the prediction to be made. Otherwise, it is quite difficult to perform it empirically, owing to a lack of resources for repeating the method for each existing hydroelectric plant. In this study, we built models using collected data from hydropower plants and the nearest meteorological stations with the aim of developing an application that would help to monitor the daily inflow one day in advance (Figure 1)[2].

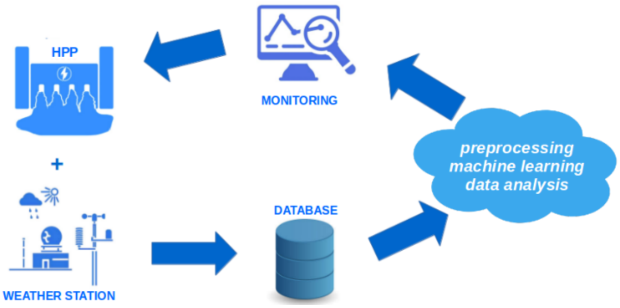


Figure 1: Graphical representation of the process: predicting inflow in the hydroelectric basins

2 METODOLOGY

2.1 Preprocessing

The daily inflow into the hydroelectric basins as labels and the amount of precipitation observed at the nearest meteorological station as descriptive features are merged by date. Then, because missing values represent less than 1% of the total data, they are filled using the average of each feature's values.

For most hydropower facilities, we could find only two descriptive features at first: the daily amount of precipitation [l] and the date. There were 11 additional available features in the data for HPP Tikvesh: time of moonrise and moonset (timestamp), intensity of precipitation $[l/m^2]$, duration of precipitation [min], time of sunrise and sunset (timestamp), the highest and lowest temperature of the day [K], absolute humidity $[g/m^3]$, cloud cover [oktas]. For HPP Tikvesh, missing values in the additional features are filled in with the average of the instances that have the same value in the most meteorologically related feature. For instance, 'humidity' and 'cloud cover'. The procedure is as follows: for a missing value in the feature 'cloud cover,' the corresponding value of 'air humidity' is X; find all the values from the feature 'cloud cover' that have the value X in the feature 'air humidity', calculate their average value and substitute for the missing value in the 'cloud cover' feature. In conjunction with the 'maximum temperature' feature, the same procedure is used to replace missing values in the 'lowest temperature' feature. Scaling the values in the descriptive features between 0 and 1 was the final step in the prepossessing process.

2.2 Feature Engineering

Using only the date and amount of precipitation as descriptive features, different regression models can be developed, but all of these predictive models have a low accuracy. Due to the computer's inability to understand the inflow pattern, including all the details that affect it, 19 new features were created from the original two. The 'date' feature was divided into three new features: day, month, and year, and they were added to the original features. Regarding the amount of precipitation and inflows into the hydroelectric basins, the derived features are the average, variance, p-variance, minimum and maximum values from the previous five days, and values from the previous day of both original features are also added as new features. Also, additional features are derived as sine and cosine functions of the days and months. The purpose of the trigonometric functions is to reduce the difference between December 31 and January 1, for instance.

Random Forest features ranking demonstrate that the features derived from the latest five days of both original features have the highest score. If we consider the correlation matrix, we can realize the same. The derived statistical features have the highest correlations with the inflow. Also it is interesting that while trigonometrically generated date features have the same correlation as the features from which they were derived, in terms of inflow, they do not correlate to each other with a degree of correlation of 1. For instance, HPP Vrben's sine function of the month, as well as the month itself (from which the sine function is derived), get a correlation index of -0.01 with the inflow, but a value of -0.04 with each other (Figure 2).

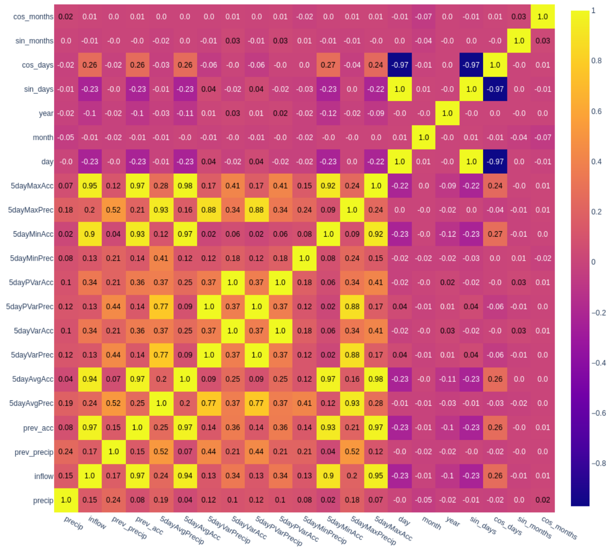


Figure 2: Correlation matrix after scaling for HPP Vrben

3 EXPERIMENTS AND RESULTS

3.1 Dataset Description

For each of the five hydropower plants in the Republic of North Macedonia that have been analyzed, the data processing and approach to the problem are the same. HPP Kozjak, HES Mavrovo power plants (consisting of HPP Vrutok and HPP Vrben), HPP Tikvesh, and HPP Shpilje are part of this study. The datasets are time-series, they were collected at daily intervals throughout an 11-year period (1/1/09 – 12/31/19) for each hydroelectric facility.

ESM[1] and UHMR[2] have collected the initial two variables, the amount of precipitation $[l/m^2]$ and inflows into the hydropower basin (Figure 3). The inflow is expressed as the maximum amount of electricity [MWh] it could be used to produce.

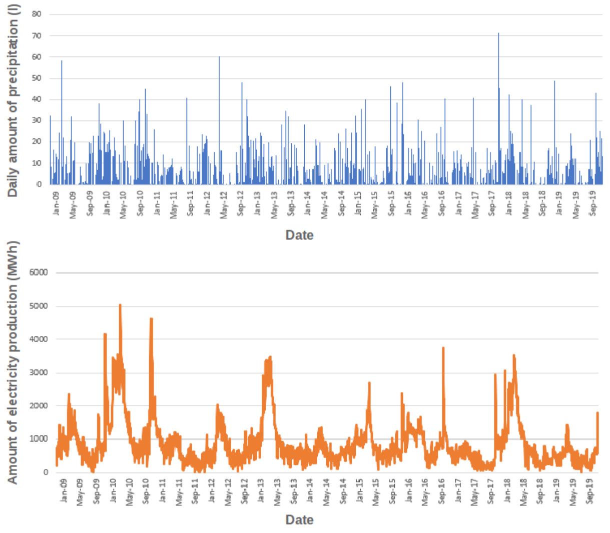


Figure 3: Top to bottom: (a) Amount of precipitation (meteorological station Debar) and (b) amount of electricity production (HPP Shpilje) during an 11-year period (2009 - 2019)

Equation 1 describes how to calculate the electricity that could be produced by hydropower plants, knowing that it is a product of power and working time [4].

$$E = \rho QgHt[Wh] \tag{1}$$

where, electricity is equal to water density ρ $[1000kg/m^3]$, multiplied by water flow (inflow) Q $[m^3/s]$, acceleration of gravity g $[m/s^2]$ and gross height drop H [m]. Inflow refers to water flowing into accumulation basins of hydropower plants. The inflow is measured in cubic meters per second $[m^3/s]$, but it can also be expressed as the quantity of electricity that the same amount of inflow could supply. When electricity is a projection of the inflow, the inflow is computed using Equation 1 and expressed in watt-hours [Wh].

3.2 Experimental Setup

To build models for one day in advance inflow prediction, we used eight different regression algorithms: Support Vector Machine, Random Forest, Linear Regression, Lasso, Gradient Boosting, Extreme Gradient Boosting, K-nearest neighbours, Decision Tree and Dummy that always predicts the mean of the training target values. Four evaluation metrics were used to evaluate the regression models for the prediction of inflow in hydropower plants: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-Squared (R2) as a standardized version of MSE.

Different numbers of selected features for each hydropower plant were considered using a class from sklearn library called SelectKBest, which selects the best features based on univariate statistical tests and the best numbers of features for each HPP were chosen based on the predictive models' errors. The experiment was divided into four parts: using all features, the best 15 features, the best 10 features, and the best 5 features, out of a total of 20 features.

Time-series can be troublesome for splits where with the shuffling process we get different train and test sets across different executions, for cross-validation, or when the test subset is before or somewhere in the middle of the train subset, etc. For instance, if a pattern appears in year 3 and persists for years 4-6, the model can detect it, even though it was not present in years 1 and 2. Because the datasets we use in this research are continuous time-series at the daily level, we split the evaluation datasets contentiously, without shuffling the subsets [8].

3.3 Results

For each of the four parts of the experiment for selecting the best n features, we calculated and plotted the mean absolute error [MWh] for each model (Figure 4, Figure 5). As we can see in Figure 4 and Figure 5, the differences in the errors are similar regardless of how many features the model is trained with. The best results are provided by different models developed using various algorithms and various number of features for each hydropower plant, but one thing that all of them have in common is that dummy regressor is the worst, while linear regression or gradient boosting produce the smallest errors. For example, if we choose HPP Vrben as one of the best outcomes, we can see that while the average daily input is 3.7 MWh, the error of the model developed using linear regression and selected 15 features is 0.34 [MWh]. Because the errors, with the exception of dummy and lasso, are not particularly big, ranging between [0.345, 0.40], all of the features listed after the five most influential features contribute a negligible percentage to improving or decreasing accuracy (Figure 5).

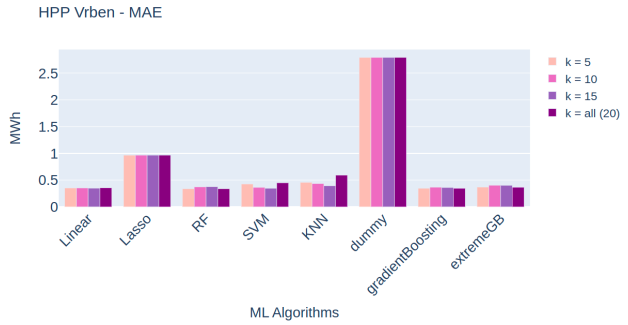


Figure 4: Graphical representation of the mean absolute error of the eight machine learning regressors used in the study, selecting different numbers of best features for HPP Vrben as a run-of-river diversion hydropower plant

4 CONCLUSIONS

Using eight different machine learning regressors, we built models for predicting inflow in Macedonian hydroelectric basins. A solution for predicting the daily inflow in hydropower plants has been proposed if the daily amount of precipitation and the amount of precipitation for the previous five days for the nearest meteorological stations are known.

[1] ESM - Elektrani na Severna Makedonija (litt. "Power plants of North Macedonia")
[2] UHMR - Uprava za Hidro-Meteoroloski raboti (National Hydrometeorological Service - Republic of North Macedonia)

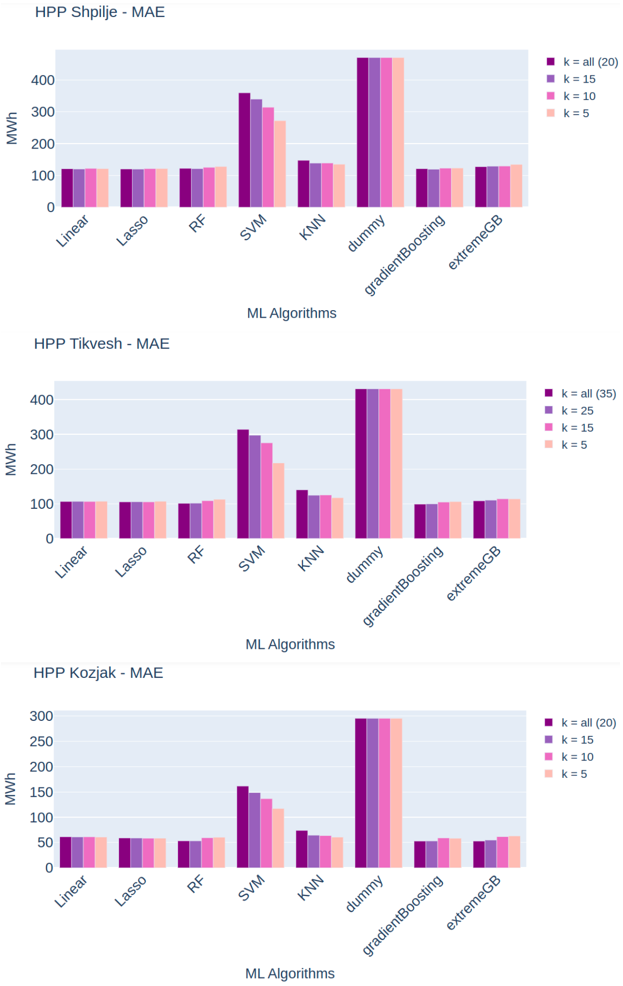


Figure 5: Top to bottom: Graphical representation of the mean absolute error of the eight machine learning regressors used in the study, selecting different numbers of best features for: HPP Shpilje, HPP Tikvesh, HPP Kozjak as storage hydropower plants

According to the results, the daily amount of precipitation and other inflow-related elements from the preceding days are the most important factors that explain inflows in hydroelectric basins. Of course, there are other variables to consider, such as temperature, cloud cover or humidity. Considering that the errors in prediction for HPP Tikvesh are not much smaller than the ones for the other HPPs where we have only the precipitation and inflow as original data and of course, the most important part - the derived features about last days, we may conclude that precipitation takes a certain amount of time to reach the basins as an inflow, depending on daily temperatures, the nature of the soil where the hydropower plant is located, and other factors.

Also, we can confirm hydrological and geological assumptions that continuity in the data is far more important in storage hydropower plants than in run-of-river diversion hydropower plants. For the first type of data, weather characteristics from previous days have no significant impact, but all original and derived attributes related to inflows and precipitation for the same day are crucial, because the rains are accumulated immediately. Otherwise, factors such as the period for melting the snow, temperature, soil moisture, and therefore the amount of inflow into the hydropower plant are significant for storage hydropower plants because it raises the river level, and thus the amount of inflow into the hydroelectric basins, but not immediately.

Because of the differences in the location, construction, and operation of hydropower plants, we can only build hydropower plant specific models. For some, the daily quantity of precipitation or the amount of precipitation from the previous day is the most essential factor, while for others, the amount of precipitation over a longer period is the most important factor. Based on this fact, which is also supported by our results for various hydropower facilities, we may conclude that we cannot build general model that can estimate the inflow for all hydropower plants. Because of the geological properties of the soil along the rivers and the temperature fluctuations in the past, we also cannot create a general model for hydropower plants of the same type.

Linear regression and gradient boosting models produce the best results. We can solve the inflow problem as a linear problem, because the relations between precipitation and inflow to the basins are simple. The precision of projected weather conditions is the key drawback for obtaining even lower errors. The more accurate the weather conditions are and the longer the time period of projected weather conditions is, the better the prediction of inflow in hydroelectric basins would be.

We built predictive models for the next day's inflow in this study. The next step is to create predictive models for as far in the future as possible, so that the model can assist in power management decisions. However, accurate projected meteorological conditions are required to develop such a model, and the further the time point, the greater the error. Hourly or minute time-series would predict more precisely in terms of time intervals.

REFERENCES

[1] Arsenov A. 2003. *Proizvodstvo na elektricna energija.* (1st. edition). ETF, Skopje.

[2] A. P. Verma A. Kusiak X. Wei and E. Roz. 2013. Modeling and prediction of rainfall using radar reflectivity data: a data-mining approach. *IEEE Transactions on Geoscience and Remote Sensing*, 2337–2342.

[3] Jianzhou Niu Tong Du Pei. ang Wendong Wang. 2019. A hybrid forecasting system based on a dual decomposition strategy and multi-objective optimization for electricity price forecasting. *Applied Energy*, 1205–1225.

[4] Djordjevic B. [n. d.] *Koriscenje vodnih snaga: Objekti hidroelektrana.* (1st. edition). Naucna knjiga, Beograd.

[5] Yazidi A. Goodwin M. 2014. A pattern recognition approach for peak prediction of electrical consumption. *IFIP Advances in Information and Communication Technology, Springer.*

[6] S. Makridakis. 2013. Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, 9 (4), 1, 527–529.

[7] Joshua S. and Coblenz. 2015. Using machine learning techniques to improve precipitation forecasting. *PLoS One.*

[8] D. Fernando S. De Silva S. Perera S. Dissanayake and W. Rankothge. 2019. Supply and demand planning of electricity power: a comprehensive solution. *IEEE Conference on Information and Communication Technology.*

[9] Stoilkov V. 2015. *Predavanja po predmetot Osnovi na obnovlivi izvori na energija.* (2nd. edition). FEEIT, Skopje.

Peak Detection for Classification of Number of Axles

Žiga Kolar
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
ziga.kolar@ijs.si

Blaž Erzar
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
blaz.erzar@gmail.com

Nika Čelan
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
nika.celan8@gmail.com

Aleksander Hrastič
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
ah3001@student.uni-lj.si

Gašper Leskovec
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
leskovecg@gmail.com

Martin Konečnik
Cestel Cestni Inženiring d.o.o
Špruha 32
Trzin, Slovenia
martin.konecnik@cestel.si

Domen Prestor
Cestel Cestni Inženiring d.o.o
Špruha 32
Trzin, Slovenia
domen.prestor@cestel.si

David Susič
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
david.susic@ijs.si

Matjaž Skobir
Cestel Cestni Inženiring d.o.o
Špruha 32
Trzin, Slovenia
matjaz.skobir@cestel.si

Matjaž Gams
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
matjaz.gams@ijs.si

ABSTRACT

A common requirement in scientific data processing is to detect peaks in a signal and to measure their positions, heights, widths, and/or areas. In this paper, the problem of peak detection from a raw signal is defined and presented. Providing the example, we showed how the problem of peak detection can be translated into detecting the number of axles in vehicles. Various algorithms for predicting the number of peaks (axles) were presented. Solution with derivatives, the solution with encoder and decoder and the solution with convolution neural network produced the best result, 99% accuracy with a certain percentage of skipped instances.

KEYWORDS

peak detection, neural networks, machine learning, signal, sensors, number of axles

1 INTRODUCTION

Identifying and analyzing peaks in a given time-series is important in many applications, because peaks are useful topological features of a time-series. In power distribution data, peaks indicate sudden high demands. In server CPU utilization data, peaks indicate sharp increase in workload. In network data, peaks correspond to bursts in traffic. In financial data, peaks indicate abrupt rise in price or volume. Troughs can be considered as inverted peaks and are equally important in many applications. Many other application areas – e.g., bioinformatics [2], mass spectrometry [4], signal processing [7, 8], image processing [10], astrophysics [13] – require peak detection.

Peak detection algorithms are also used for classification of the number of peaks or axles. For example, when a vehicle places one of its tyres on a weight sensor, a peak is detected in the signal. Each peak represents one vehicle axle. Therefore, the algorithm detecting how many peaks occur in a given signal in this way detects the number of axes. For the purpose of this study, 16 different signals for two driving lanes were provided by company Cestel. Sensors were placed under a bridge near Obrežje. Sensors 1 and 2 were placed at the beginning and end of measuring area for lane 1. Sensors 15 and 16 were placed on lane 2 in a similar fashion. The rest were placed perpendicular on the road between the pairs. The main goal of this paper is to predict the number of axles as accurately as possible with the use of mathematical models and machine learning algorithms given signals. We introduce the solution using deep neural networks (artificial neural network and convolution neural network), regular derivatives, predefined library find_peaks and a package $tsfresh$ for peak detection. In theory, peak detection is formally a trivial task, however, in reality the task can be performed only with some degree of accuracy.

The rest of the paper is organized as follows. Section 2 presents related work. Main methodology and algorithms are described in section 3. Finally, section 4 concludes the paper with summary and ideas for future work.

2 RELATED WORK

Peak detection is a common task in time-series analysis and signal processing. Standard approaches to peak detection include (i) using smoothing and then fitting a known function (e.g., a polynomial) to the time-series; and (ii) matching a known peak shape to the time-series. Another common approach to peak-trough detection is to detect zero-crossings (i.e., local maxima) in the

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

differences (slope sign change) between a point and its neighbours. However, this detects all peaks-troughs, whether strong or not. To reduce the effects of noise, it is required that the local signal-to-noise ratio (SNR) should be over a certain threshold [8, 11]. The key question now is how to set the correct threshold so as to minimize false positives. Ma, van Genderen and Beukelman et al. [10] compute the threshold automatically by adapting it to the noise levels in the time-series as $h = \frac{max+abs_{avg}}{2} + K * abs_{dev}$, where max is the maximum value in the time-series, abs_{avg} is the average of the absolute values in the time-series, abs_{dev} is the mean absolute deviation and K is a user-specified constant. Azzini et al. [2] analyze peaks in gene expression microarray time-series data (for malaria parasite Plasmodium falciparum) using multiple methods; each method assigns a score to every point in the time-series. In one method, the score is the rate of change (i.e., the derivative) computed at each point. In another method, the score is computed as the fraction of the area under the candidate peak. Top 10 candidate peaks are selected for each method; peaks detected by multiple methods are chosen as true peaks. The detected peaks are used to identify genes; SVM are then used to assign a functional group to each identified gene. Key problems in peak detection are noise in the data and the fact that peaks occur with different amplitudes (strong and weak peaks) and at different scales, which result in a large number of false positives among detected peaks. Based on the observation that peaks in mass spectroscopy data have characteristic shapes, Du, Kibbe and Lin et al. [5] propose a continuous wavelet transform (CWT) based pattern-matching algorithm for peak detection. 2D array of CWT coefficients is computed (using a Mexican Hat mother wavelet which has the basic shape like a peak) for the time-series at multiple scales and ridges in this wavelet space representation are systematically examined to identify peaks. Coombes et al. [4] and Lange et al. [9] present other approaches for peak detection using wavelets and their applications to analyze spectroscopy data. Zhu and Shasha et al. [13] propose a wavelet-based burst (not peak) detection algorithm. The wavelet coefficients (as well as window statistics such as averages) for Haar wavelets are organized in a special data structure called the shifted wavelet tree (SWT). Each level in the tree corresponds to a resolution or time scale and each node corresponds to a window. By automatically scanning windows of different sizes and different time resolutions, the bursts can be elastically detected (appropriate window size is automatically decided). Zhu and Shasha et al. [13] apply their technique to detecting Gamma Ray bursts in real-time in the Milagro astronomical telescope, which vary widely in their strength and duration (from minutes to days). Harmer et al. [7] propose a momentum-based algorithm to detect peaks. The idea is compute velocity (i.e., rate of change) and momentum (i.e., product of value and velocity) at various points. A “ball” dropped from a previously detected peak will gain momentum as it climbs down and lose momentum as it climbs the next peak; the point where it comes to rest (loses all its momentum) is the next peak. Simple analogs of the laws in Newtonian mechanics are proposed (e.g., friction) to compute changes in momentum as the ball traverses the time-series.

3 METHODOLOGY

In this section, algorithms for peak detection are described. Each machine learning method uses 62076 samples (vehicles) and up to 16 signals for classification of number of axles. Typically, only the first signal was chosen for training the model. This is because the first signal had less noise than other signals. Each signal has different length. Therefore, the signals that had length less that maximum time had to be extended to maximum signal time in order to create features with the same length. To achieve this, additional zeros were filled to the positions up to the maximum signal time. Maximum time is 6113, which is equal to the number of features. Figure 1 shows a signal from Sensor 1 which has maximum sensor time.

Each method uses classification accuracy for evaluation of the model. Classification accuracy is a metric that summarizes the performance of a classification model as the number of correct predictions divided by the total number of predictions. In this study, correct predictions are correctly predicted peaks(number of axles).

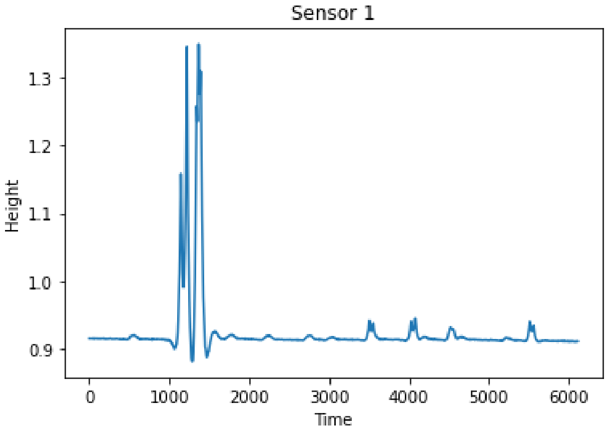


Figure 1: Signal with maximum sensor time.

3.1 Peak Detection with Derivatives

Since peaks are local maxima, we can use mathematical methods for finding them. But because we are not working with functions but rather noisy discrete signals, we need to modify them slightly.

Let us define s_i as i^{th} signal value. First we remove noise at the beginning and the end of the signal where there are no axes. To do so we find the horizontal line with maximum support, where support for line at height y is defined as the number of signal values s_i for which $|y - s_i| < margin$. For line height y we take n equally spaced values from interval $[min, max]$ and half the distance between consecutive y values is used as $margin$. Here min and max represent minimum and maximum value of the signal. To remove noise and normalize signal, we now define:

$$\hat{s}_i = \begin{cases} 0, & \text{if } s_i < y + 2 * margin. \\ \frac{s_i - min}{max - min}, & \text{otherwise.} \end{cases}$$

On signal $\hat{s}$ we calculate first and second derivative - $\dot{s}$ and $\ddot{s}$ - using finite difference, which is implemented using convolution with kernels $[-0.5, 0, 0.5]$ and $[1, -2, 1]$. Finally peaks can be acquired by finding indices i for which $\hat{s}_i > 0.25$, $|\dot{s}_i| < 0.01$ and $|\ddot{s}_i| < 0$ while only taking peaks which are local minima, i.e. $s_i > \max\{s_{i-1}, s_{i+1}\}$.

This procedure achieves accuracy ≈ 90% when using sensor 1. When using other sensors, accuracy is lower, but it can be improved by choosing the correct sensor for every instance. We do this by training nine models: one regression model M_a for predicting number of axes and eight models M_k, $k = 1..8$, for predicting whether prediction on sensor k is correct. We only use sensors from lane 1, since sensors on the other lane give poor accuracy.

First we define $p^{(i)}$ as correct number of peaks for instance i and $p_k^{(i)}$ as number of peaks detected by procedure described in this section for instance i using sensor k. Now we create matrix X and vector y on which we train gradient boosting regression. This model predicts number of axes from number of peaks detected on all sensors. Matrix X contains one row $x^{(i)} = [p_1^{(i)}, p_2^{(i)}, \ldots, p_8^{(i)}]$ for each instance i with one column for every sensor, while vector y contains ground truth values for number of axes:

$$X = \left[x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right]^T, \quad y = \left[p^{(1)}, p^{(2)}, \ldots, p^{(m)}\right]^T.$$

Here m is number of all instances. Other eight models use the same matrix X, but different vector y. Model M_k which predicts whether detection using sensor k produces correct number of peaks uses:

$$y_k = \left[p_k^{(1)} = p^{(1)}, p_k^{(2)} = p^{(2)}, \ldots, p_k^{(m)} = p^{(m)}\right]^T,$$

where equality comparison evaluates to 1 when true and 0 when false. Gradient boosting classifiers are used for these models.

After all nine models are trained, peaks on a new instance $m+1$ can be detected by first using the described peak detection procedure on all eight sensors to obtain input vector $x^{(m+1)}$:

$$x^{(m+1)} = [p_1^{(m+1)}, p_2^{(m+1)}, \ldots, p_8^{(m+1)}].$$

This vector is then first fed into M_a model to predict number of axes and the result is rounded to closest integer value to get a. Furthermore models M_k are used to get confidence c_k for each sensor. Now valid sensors are the ones using which correct number of peaks were detected and have confidence higher than some threshold T:

$$sensors = \{k \mid 1 \leq k \leq 8 \wedge p_k^{(m+1)} = a \wedge c_k > T\}.$$

If $sensors = \emptyset$, instance $m+1$ is skipped, otherwise a axes are predicted and $\min\{sensors\}$is the best sensor for detection. For $T = 0.95$ this system has accuracy 99.5% while skipping 20% of instances.

3.2 Peak Detection with Encoder/Decoder

Since we know where peaks are located in every signal, we can train a model that will for every instance predict locations of peaks. Because we are working with time series data, we can use a one dimensional convolutional neural network with autoencoder architecture. This allows us to predict locations for variable number of peaks. Inputs and outputs have the same dimensions, while the model consists of two parts: encoder, to create low dimensional embedding in latent space, and decoder, to reconstruct output from it.

As inputs we use signals from sensor 1. On output we want to predict a vector of the same dimension, which has ones in time slots containing a peak and zeros everywhere else. Because CNNs take inputs of the same length, we pad all input and output vectors to maximum length. To make maximum length smaller, we use the noise removal method from section 3.1 to crop noise at the beginning and at the end from the signals.

Encoder is made of 3 convolutional layers. Each is followed by batch normalization and max pooling of size 2. Convolutional layers use ReLU activation, 8, 16 and 32 filters and sizes 5, 2 and 3 respectively. Decoder has the same structure with number of filters and sizes reversed and max pooling layers replaced with up sampling. This model is then trained using Adam optimizer and binary cross entropy loss function.

After model is trained, peaks on new instance can be detected by feeding sensor 1 signal s to it to obtain prediction vector p. Peaks are now located at indices for which prediction value is a strong enough local maximum and signal amplitude is high enough:

$$peaks = \{i \mid p_i \geq \max\{p_{i-5:i+5}\} \wedge \hat{p}_i > T_1 \wedge \hat{s}_i > 0.15\},$$

while skipping instances for which $\max\{p_i \mid i \in peaks\} < T_2$. Here $\hat{p}, \hat{s}$ are normalized to contain values in $[0, 1]$ and T_1, T_2 are thresholds that need to be selected. For $T_1 = 0.01$ and $T_2 = 0.5$ accuracy 99.6% is achieved with 20% skipped instances.

3.3 Peak Detection with Artificial Neural Network

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another [12].

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network [12].

In this method, artificial neural network was used to predict the number of peaks. Neural networks were a viable solution for this problem, because we had enough data at our disposal for deep learning. Whole signal from sensor 1 was provided as input layer. Architecture of the neural network contains two hidden layers, with 16 and 12 neurons, respectively. Output data (number of peaks) was one-hot encoded, therefore softmax activation function was used in the output layer. Model returned the probability for each class. In the end, an algorithm picked the column with the highest probability (i-th column depicts i number of peaks). This model achieved 91% accuracy for predicting the number of peaks.

3.4 Peak Detection with Convolution Neural Network

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- Convolutional layers which convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field.
- Pooling layers which reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer.
- Fully-connected layers which connect every neuron in one layer to every neuron in another layer.

With each layer, the CNN increases in its complexity, identifying greater portions of the required information [1].

Convolutional neural network was utilized to predict the number of peaks. The motivation for the usage of this type of network comes from a fact that convolutional neural networks work well with time series data. Whole signal from sensor 1 was used as input layer. Architecture of the network contains three 1D convolution layers and three 1D pooling layers. At the end we used the fully connected layer with 100 neurons. Output layer has a softmax activation layer. Similarly than in subsection 3.3, the model returned the probability for each class and in the end, algorithm picked the column with the highest probability. This model achieved 97% accuracy. If we decide to skip 6.5% samples that are below the 99% probability threshold, we achieve the accuracy of 99.1%.

3.5 Peak Detection with Predefined Method Find_peaks

Another method for peak detection is by using the predefined function named *find_peaks*. This function takes a 1-D array and finds all local maxima by simple comparison of neighboring values. In the context of this function, a peak or local maximum is defined as any sample whose two direct neighbours have a smaller amplitude [6]. Because each signal has different maximum height, the parameter in function find_peaks named *height* differs from sample to sample. Height parameter is defined as minimal required height for peaks to be detected. Peaks below that threshold are not detected. Height was calculated by formula: height = | |max(sensorHeight)| - |min(sensorHeight)| | / 10. The above described method achieved 89% accuracy and also returned the position of every peak. An example can be seen on Figure 2 on which 2 peaks are detected. They are marked with 2 oranges crosses.

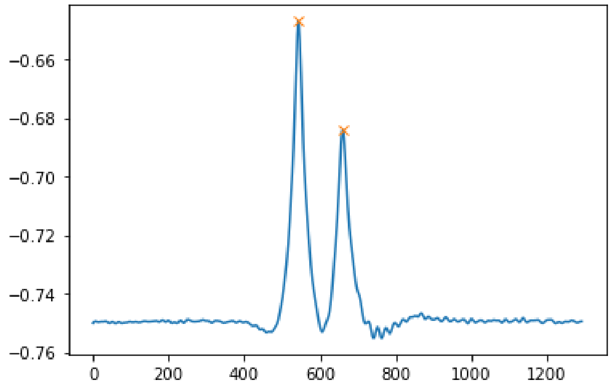


Figure 2: Signal with two peaks. Location of the peak is marked with an orange cross.

3.6 Peak Detection with Library Tsfresh

Another alternative approach for peak prediction is by using the python package named *tsfresh*. It automatically calculates a large number of time series characteristics, the so called features. Furthermore, the package contains methods to evaluate the explaining power and importance of such characteristics for regression or classification tasks. *tsfresh* is used for systematic feature engineering from time-series and other sequential data. These data have in common that they are ordered by an independent variable. The most common independent variable is time (time series) [3]. After the features were extracted, they were used by gradient boost classifier for predicting the number of peaks. This approach produced 89% accuracy predicting the number of peaks.

4 CONCLUSION AND DISCUSSION

We defined and presented the problem of peak detection from a raw signal. Providing the example, we showed how the problem of peak detection can be translated into detecting the number of axles in vehicles. Various algorithms for predicting the number of peaks (axles) were presented. The solution with derivative, the solution with encoder and decoder and the solution with convolution neural network produced the best results, 99% accuracy with a certain percentage of skipped instances. In future work, the mentioned results can be tweaked and improved by using different learning parameters, e.g. different learning rate, different number of neurons, different activation function. Furthermore, better results can be achieved by changing the architecture of the neural network, e.g. different or more convolution or pooling layers. The results of peak detection can also be extended into determining the axle distances. Once the axle number is accurately predicted, a new set of algorithms can be implemented to solve this new task.

ACKNOWLEDGMENTS

This study received funding from company Cestel. The authors acknowledge the funding from the Slovenian Research Agency (ARRS), Grant (PR-10495) and Basic core funding P2-0209.

REFERENCES

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*. Ieee, 1–6.

[2] Ivano Azzini, Rossana Dell'Anna, Federica Ciocchetta, Francesca Demichelis, Andrea Sboner, Enrico Blanzieri, and A Malossini. 2004. Simple methods for peak detection in time series microarray data. *Proc. CAMDA'04 (Critical Assessment of Microarray Data)*.

[3] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. 2018. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307, 72–77.

[4] Kevin R Coombes, Spiridon Tsavachidis, Jeffrey S Morris, Keith A Baggerly, Mien-Chie Hung, and Henry M Kuerer. 2005. Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics*, 5, 16, 4107–4117.

[5] Pan Du, Warren A Kibbe, and Simon M Lin. 2006. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *bioinformatics*, 22, 17, 2059–2065.

[6] 2022. Find peaks. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html. (2022).

[7] Karl Harmer, Gareth Howells, Weiguo Sheng, Michael Fairhurst, and Farzin Deravi. 2008. A peak-trough detection algorithm based on momentum. In *2008 Congress on Image and Signal Processing*. Vol. 4. IEEE, 454–458.

[8] Valentin T Jordanov and Dave L Hall. 2002. Digital peak detector with noise threshold. In *2002 IEEE Nuclear Science Symposium Conference Record*. Vol. 1. IEEE, 140–142.

[9] Eva Lange, Clemens Gröpl, Knut Reinert, Oliver Kohlbacher, and Andreas Hildebrandt. 2006. High-accuracy peak picking of proteomics data using wavelet techniques. In *Biocomputing 2006*. World Scientific, 243–254.

[10] Meng Ma, Arjan Van Genderen, and Peter Beukelman. 2005. Developing and implementing peak detection for real-time image registration. In *Proceedings of the 16th Annual Workshop on Circuits, Systems & Signal Processing (ProRISC2005)*. Citeseer, 641–652.

[11] GM Nijm, AV Sahakian, S Swiryn, and AC Larson. 2007. Comparison of signal peak detection algorithms for self-gated cardiac cine mri. In *2007 Computers in Cardiology*. IEEE, 407–410.

[12] Sun-Chong Wang. 2003. Artificial neural network. In *Interdisciplinary computing in java programming*. Springer, 81–100.

[13] Yunyue Zhu and Dennis Shasha. 2003. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 336–345.

Unified Question Answering in Slovene

Katja Logar and Marko Robnik-Šikonja
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia
kl2164@student.uni-lj.si,marko.robnik@fri.uni-lj.si

ABSTRACT

Question answering is one of the most challenging tasks in language understanding. Most approaches are developed for English, while less-resourced languages are much less researched. We adapt a successful English question-answering approach, called UnifiedQA, to the less-resourced Slovene language. Our adaptation uses the encoder-decoder transformer SloT5 and mT5 models to handle four question-answering formats: yes/no, multiple-choice, abstractive, and extractive. We use existing Slovene adaptations of four datasets, and machine translate the MCTest dataset. We show that a general model can answer questions in different formats at least as well as specialized models. The results are further improved using cross-lingual transfer from English. While we produce state-of-the-art results for Slovene, the performance still lags behind English.

KEYWORDS

question answering, Slovene language, deep neural networks, encoder-decoder models, natural language processing

1 INTRODUCTION

Most studies for the question answering (QA) task deal with the English language. This leaves many language specifics, not present in English, potentially inadequately addressed. E.g., some problematic language specifics in morphologically-rich Slovene language are noun and adverb declension, three different genders, three counts, the person or pronoun being hidden in a verb, etc. An additional problem for less-resourced languages is the lack of suitable datasets for QA.

Khashabi et al. [5] argue that building specialized models for each QA dataset or QA format is unnecessary, as they all require a similar inference capability. Therefore, it is possible to develop one model capable of answering questions in different formats. They call their approach UnifiedQA, and we adapted this approach to Slovene.

The number of QA datasets in Slovene is much lower than used in the original UnifiedQA. We found four partially human-translated but mostly machine-translated datasets. To improve that, we first machine translate the additional MCTest dataset [9] into Slovene and fix translation errors.

Our method is based on the pretrained Slovene encoder-decoder transformer model SloT5 [11]. We finetune the model on the five QA datasets and analyze its performance. We also test the role of uppercase and lowercase letters, the impact of unanswerable questions, and the contribution of each dataset to the performance of the unified model. Next, we test the cross-lingual transfer and train a multilingual question answering model based on the multilingual mT5 model [13], using English and Slovene datasets. Finally, we perform a qualitative analysis of the obtained models. The results show that our system is currently the best performing QA system for Slovene. We make its source code freely accessible[1].

The paper is split into four further sections. In Section 2, we outline the related work on QA in Slovene. Section 3 presents our adaptation of UnifiedQA methodology and the applied Slovene QA datasets, and Section 4 discusses different evaluation settings and their results. In Section 5, we present the findings and ideas for further improvements.

2 RELATED WORK

The QA in Slovene is relatively unexplored. In the pre-neural setting, Čeh et al. [1] developed a closed-domain QA system for answering common questions that arise during students' studies at the University of Maribor, Faculty of Electrical Engineering, Computer Science and Informatics. The translation of the SuperGLUE benchmark suite to Slovene in 2021 [14] provided four partially human, partially machine translated QA datasets (BoolQ, COPA, MultiRC, and ReCoRD) and evaluation of Slovene BERT models. Ulčar et al. [11] adapted the SloT5 model for the yes-no and multiple-choice questions. Finally, Zupanič et al. [15] translated the SQuAD 2.0 dataset from English and adapted different multilingual models. They achieved the best result with the SloBERTa 2.0 model [12]. In contrast to the above works, we apply the transfer learning paradigm within the encoder-decoder SloT5 and mT5 models and provide a unified approach to different QA formats, obtaining the best results so far.

3 METHODOLOGY

Our methodology follows Khashabi et al. [5] UnifiedQA methodology. The authors define four QA formats (extractive, abstractive, multiple-choice, and yes/no) and unify the learning approach to these formats. The extractive format requires that the answer is directly stated in the supplied context as a substring. The abstractive format requires paraphrasing of the given context and the answer may require linking information from several sentences. The multiple-choice datasets have possible answers listed and the aim is to select the given option correctly. Finally, the yes/no questions require only yes or no as an answer.

The datasets with different QA formats are converted to text format, with parts of the input separated by the `"\n"` separator. Extractive, abstractive and yes/no questions are coded as `"question \n context"` and multiple-choice questions as `"question \n possible choices \n context"`. Here, the possible choices are indicated in capital letters from A onwards `(A) choice 1 (B) choice 2...`.

We initially considered four QA datasets. Three stem from the translation of the SuperGLUE benchmark to Slovene [14]: MultiRC [4] (abstractive), COPA [10] (multiple-choice) and BoolQ [2] (yes/no). We also used the SQuAD 2.0 [8] (extractive) Slovene

Information Society 2022, 11–14 October 2022, Ljubljana, Slovenia

[1] https://github.com/klogar/QAslovene

translation [15]. SQuAD 2.0 contains unanswerable questions, and some are also present in MultiRC. As we focus on the reading comprehension task, all selected datasets have a context. COPA is a commonsense reasoning dataset, which is not our primary focus, but we included it due to being human translated into Slovene. BoolQ, MultiRC, and SQuAD 2.0 are partially human translated [14, 15].

To have a non-commonsense multiple-choice dataset, we machine translated the MCTest dataset [9] and fixed some translation errors. To reduce the cost of translation, we partially used the commercial solution DeepL [3] and partially an internal neural machine translator of a bit lesser quality. Later, we translated the entire MCTest dataset with the DeepL translator and made it publicly available in our repository. However, the reported results are obtained using the initial mixed translation setting.

As the starting training model for monolingual Slovene UnifiedQA models, we used the monolingual Slovene variant of the T5 transformer encoder-decoder model [7], called SloT5 [11]. For the cross-lingual transfer experiments, we applied the multilingual variant of T5, called mT5 [13]. Due to computational time and GPU memory limitations, we used the SloT5 and mT5 models of the smallest size (60M and 300M parameters, respectively). Originally, Khashabi et al. used the T5 model [7] of the largest possible size (11B parameters) and the BART$_{\text{large}}$ model [6] as a starting point for the UnifiedQA model. However, they also report results for the T5$_{\text{small}}$ model, which we report for comparison, so all models are of comparable sizes. Table 1 lists the parameters used to finetune our models.

Table 1: Parameters for finetuning UnifiedQA models.

Parameter	Value
Maximum input size [tokens]	512
Maximum output size [tokens]	100
Number of epochs	25
Batch size	8
Number of beams	4
Learning rate	5e-5

4 EXPERIMENTS AND RESULTS

In this section, we report our work on empirical evaluation. We present the evaluation metrics, original English results, experiments and results in the monolingual Slovene setting, and in the cross-lingual transfer setting.

4.1 Evaluation Metrics

For each dataset, we use a different evaluation metric. For BoolQ, we report the classification accuracy; for SQuAD 2.0, the F_1 score; for MultiRC, we use ROUGE-L; and for the multiple-choice datasets (MCTest and COPA), we calculate the best match between the generated text and the offered options and compute the classification accuracy. In all cases, the answers are first normalized (removing punctuation and unnecessary spaces and converting the text to lowercase).

4.2 English UnifiedQA Results Using T5$_{\text{small}}$

First, we replicated the results of the original English UnifiedQA [5] and also obtained the results for the datasets not originally used, i.e. COPA and MultiRC (the latter was only used as a yes/no dataset in [5]). The results are presented in Table 2. The results for BoolQ and MCTest are slightly worse than originally reported, which could be attributed to slightly different parameters for text generation. We achieved a much worse result for the SQuAD 2.0 dataset, with F_1 only 46.1% rather than 67.6%. Trying to replicate the published scores with the original code[2], we obtained similar results to ours . However, we analyzed the difference and believe that at least some of them are due to unanswerable questions, as the F_1 score is 84.5% for questions that have an answer and only 7.8% for unanswerable questions. The UnifiedQA model, therefore, does a poor job of detecting if a question is unanswerable from the context.

Table 2: Our and published results of the UnifiedQA (UniQA) approach on English datasets using the T5$_{\text{small}}$ model.

Dataset Metric	BoolQ CA	COPA CA	MCTest CA	MultiRC ROUGE-L	SQuAD 2.0 F_1
UniQA(publ.)	0.771	/	0.800	/	0.676
UniQA(ours)	0.757	0.560	0.762	0.536	0.461

4.3 Slovene Monolingual Results Using SloT5

In the Slovene monolingual setting, we compare different variants of Slovene UnifiedQA models and report the results in Table 3. We adapted the models for each QA format separately and obtained so-called specialized models. These provided a baseline for what could be achieved with each individual QA format. We then trained the SloUnifiedQA model using all available Slovene datasets. We also investigated the impact of unanswerable questions (SloUnifiedQA-NA, SloUnifiedQA-NA2, explained below) and the use of only lower case letters (SloUnifiedQA-LC).

Table 3: Comparing variants of Slovene UnifiedQA approach (based on the SloT5 model). Besides the unified model, we report the results of specialized models for each QA format (specialized), the best results published so far on these datasets (published), and the default classifier. The effect of unanswerable questions and lowercasing is analyzed in the bottom part of the table. Note that SloUniQA-NA is tested on modified datasets without unanswerable questions, so the results for this model are incomparable.

Dataset Metric	BoolQ CA	COPA CA	MCTest CA	MultiRC ROUGE-L	SQuAD 2.0 F_1	Avg.
SloUniQA	0.683	0.532	0.463	0.310	0.555	0.509
specialized	0.688	0.486	0.439	0.255	0.554	0.484
published	0.666	0.500	/	/	0.739	/
default	0.623	0.500	0.269	/	/	/
SloUniQA-NA	0.675	0.524	0.454	0.319	0.637	0.522
SloUniQA-NA2	**0.695**	**0.554**	**0.474**	**0.321**	**0.556**	**0.520**
SloUniQA-LC	0.686	0.530	0.449	0.259	0.533	0.491

Comparing the SloUnifiedQA model with specialized models, the models achieve better results for the multiple-choice datasets (COPA and MCTest) and the abstractive dataset (MultiRC). The improvement for the extractive dataset is minimal, and we observe a slight decrease in accuracy for the yes/no dataset (BoolQ). Better results are also obtained compared to all main classifiers.

[2] https://github.com/allenai/unifiedqa

Comparing SloUnifiedQA on Slovene with the English UnifiedQA model on English datasets (in Table 2), the English model gives better results for all selected formats except SQuAD 2.0. Interestingly, the English and Slovene models have different problems with SQuAD 2.0. The Slovenian one predicts unanswerable questions too often (it has F_1 score of 60,3% for unanswerable questions and only 50,4% for answerable ones, while incorrectly identifying 13% of answerable questions as unanswerable), the English one too rarely. At the same time, the English model never wrongly predicts that a question is unanswerable. This is likely due to unanswerable questions making up a larger proportion of the dataset in the Slovene training dataset than in the English one. For other datasets, the biggest difference in metrics can be observed in the MCTest multiple-choice dataset, where the difference is 33%. We attribute the worse result of SloUnifiedQA to machine translations and a much smaller training dataset, especially for the multiple-choice questions; as in the original work, the authors use three additional datasets in addition to MCTest.

Compared to other published works on the same datasets, we achieve better results with the SloUnifiedQA on the BoolQ and COPA datasets compared to Ulčar and Robnik-Šikonja [11], while on the SQuAD 2.0 dataset, Zupanič et al. [15] achieve a significantly better result (almost 20%). Here, Ulčar and Robnik-Šikonja [11] also use the SloT5 model with the textual output, while Zupanič et al. [15] use the SloBERTa model and only predict the span of the answer, which is an easier task.

4.3.1 *The Effect of Unanswerable Questions.*

Unanswerable questions account for about one-third of all training examples, and models could overfit such questions. To address this issue, we train two models, SloUnifiedQA-NA and SloUnifiedQA-NA2. For the SloUnifiedQA-NA model, we removed all unanswerable questions. As evident from Table 3, for yes/no questions and multiple-choice questions the accuracy deteriorates, while for abstractive and extractive questions the metrics improve. The biggest improvement occurred for the SQuAD 2.0 dataset, where the F_1 metric for answerable questions improved to 63.7%.

The SloUnifiedQA-NA was the basis for the SloUnifiedQA-NA2 model, which we trained on complete datasets, including unanswerable questions. The metrics slightly improved for BoolQ, COPA, and MCTest but may be due to the longer training time. No improvement is observed for SQuAD 2.0; the F_1 for answerable questions even drops to 51.5%.

4.3.2 *The Effect of Using Lower Case Letters.*

To analyze the effect of using only lower case letters, we trained the SloUnifiedQA-LC model. The results are comparable for BoolQ and COPA, but for MCTest, MultiRC, and SQuAD 2.0, the results are worse. The uppercase letters, therefore, contain relevant information in Slovene.

4.3.3 *Contribution of Datasets in the Unified Model.*

To assess the impact of each dataset in the SloUnifiedQA model, we dropped each training dataset in turn. The results are shown in Table 4. The largest individual performance drop is observed for the model without BoolQ, as the yes/no questions become unanswerable (the CA for the BoolQ dataset is almost 0%). This also strongly affects the average impact but causes even slight improvements on MCTest, MultiRC, and SQuAD 2.0. The second largest average performance drop is achieved by the model without SQuAD 2.0, where a drop is observed on all datasets. For other models, the drops are observed mainly on datasets on which models were not trained. Overall, the COPA dataset contributes the least to the performance of SloUnifiedQA, the corresponding model achieving almost the same performance.

Table 4: Contribution of datasets in the unified model by omitting one dataset at a time. The red color indicates the two largest performance drops for each dataset.

Dataset Metric	BoolQ CA	COPA CA	MCTest CA	MultiRC ROUGE-L	SQuAD2.0 F_1	 Avg.
SloUniQA	0.683	0.532	0.463	0.310	0.555	0.509
no BoolQ	0.001	0.522	0.486	0.319	0.561	0.378
no SQuAD 2.0	0.664	0.516	0.451	0.258	0.120	0.402
no MCTest	0.676	0.510	0.351	0.317	0.560	0.483
no MultiRC	0.690	0.536	0.457	0.209	0.552	0.489
no COPA	0.683	0.510	0.456	0.319	0.554	0.504

4.4 Cross-Lingual Transfer Using mT5

There are only a few QA datasets in Slovene, so we checked if using transfer from additional English datasets can improve the Slovene results. We used three different collections of datasets.

- **SLO**: Slovene datasets BoolQ, COPA, MCTest, MultiRC and SQuAD 2.0 (described in Section 3).
- **ANG5**: English datasets BoolQ, COPA, MCTest, MultiRC, and SQuAD 2.0 (the English dataset, whose translations form the SLO collection).
- **ANG9**: English datasets BoolQ, COPA, MCTest, MultiRC, and SQuAD 2.0 and all datasets, used by Khashabi et al. [5], except SQuAD 1.1, i.e. NarrativeQA, RACE, ARC, and OBQA.

We trained five models using the multilingual mT5 model on these dataset collections and tested them on the SLO test sets. The first model, mSloUnifiedQA, was trained only on SLO datasets and gives a baseline performance of mT5, also enabling comparison to monolingual SloT5. The mSloUnifiedQA$_1$ models were trained on both English and Slovene datasets simultaneously (only one phase), with the English dataset collection being either ANG5 or ANG9. Only the SLO dataset group was used for validation. The mSloUnifiedQA$_2$ models were trained in two phases, first on the English datasets (ANG5 or ANG9), using the ROUGE-L metric to select the best model, and the obtained model was then finetuned on the SLO dataset collection.

The results of the five multilingual models are presented in Table 5. Comparison between the monolingual SloUnifiedQA model (in Table 3) and the multilingual mSloUnifiedQA shows that they perform on average equally well, with SloUnifiedQA performing better on the BoolQ, COPA and MultiRC datasets, and mSloUnifiedQA performing better on the MCTest and SQuAD 2.0 datasets.

Adding additional knowledge in English improved the average metrics by 3-4%, but the training time increased by about

Table 5: Results of cross-lingual transfer using additional English datasets and multilingual models based on mT5.

Dataset Meric	BoolQ CA	COPA CA	MCTest CA	MultiRC ROUGE-L	SQuAD 2.0 F_1	 Avg.
mSloUniQA	0.646	0.488	0.515	0.298	0.571	0.504
mSloUniQA$_1$ (ANG5)	0.672	0.486	0.582	0.308	0.587	0.527
mSloUniQA$_1$ (ANG9)	0.676	0.508	0.579	**0.340**	0.598	**0.540**
mSloUniQA$_2$ (ANG5)	0.682	0.504	0.564	0.313	0.593	0.531
mSloUniQA$_2$ (ANG9)	**0.683**	0.486	**0.602**	0.323	**0.604**	**0.540**

four times for the models with the most datasets (ANG9). A slight improvement can be observed for models using nine English datasets (ANG9) relative to those with only five English datasets (ANG5). The additional datasets contribute the most to the MCTest multiple-choice results, but the performance on MultiRC and SQuAD 2.0 also improved. On the other hand, despite the additional datasets, the results for BoolQ and COPA are worse than for the monolingual model. Using one or two-phase training does not make a difference on average, but there are differences in individual datasets.

4.5 Qualitative Analysis

Qualitative analysis of our models showed that the generated answers are mostly substrings or given choices in multiple-choice questions. Models cannot paraphrase, rephrase or provide answers in the correct Slovene case. They also have problems with multi-part questions requiring multiple answers that are not listed in the same place in the context. Machine translations, which are not always grammatically correct or do not make it clear what the question is asking for, also make answering the questions difficult. The models performed best on factoid questions that require a short answer.

5 CONCLUSION AND FUTURE WORK

The main contributions of this work are the generative unified QA models based on SloT5 and mT5 encoder-decoder transformer models, which set new state-of-the-art results for QA in Slovene. An additional contribution is the machine-translated and corrected MCTest dataset.

We identify three possible directions for further work. First, better translations or dedicated Slovenian datasets would improve upon currently mainly machine-translated datasets. Second, larger T5 models and longer training times have shown better performance in English. In our work, we used only the smallest available T5 models due to the limited memory of the GPU; we also limited training sessions to a maximum of 25 epochs. Third, by using new datasets, especially additional multiple-choice datasets, as evidenced by the improvement brought by the introduction of English multiple-choice datasets. Further, additional abstractive datasets could teach the models to rephrase better or that answers shall not be just substrings of the provided context.

ACKNOWLEDGMENTS

Marko Robnik-Šikonja received financial support from the Slovenian Research Agency through core research programme P6-0411 and projects J6-2581 and J7-3159, as well as the Ministry of Culture of Republic of Slovenia through the project Development of Slovene in Digital Environment (RSDO).

REFERENCES

[1] Ines Čeh and Milan Ojsteršek. “Slovene language question answering system”. In: *Recent advances in computers: Proceedings of the 13th WSEAS international conference on computers (part of the 13th WSEAS CSCC multiconference).* 2009, pp. 502–508.

[2] Christopher Clark et al. “BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* 2019, pp. 2924–2936.

[3] *DeepL Translator.* [18 July 2022]. URL: https://www.deepl.com/translator.

[4] Daniel Khashabi et al. “Looking beyond the surface: A challenge set for reading comprehension over multiple sentences”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* 2018, pp. 252–262.

[5] Daniel Khashabi et al. “UnifiedQA: Crossing Format Boundaries With a Single QA System”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings.* 2020, pp. 1896–1907.

[6] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* 2020, pp. 7871–7880.

[7] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21 (2020), pp. 1–67.

[8] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* 2018, pp. 784–789.

[9] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. “MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* 2013, pp. 193–203.

[10] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. “Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning.” In: *AAAI spring symposium: logical formalizations of commonsense reasoning.* 2011, pp. 90–95.

[11] Matej Ulčar and Marko Robnik-Šikonja. “Sequence to sequence pretraining for a less-resourced Slovenian language”. In: *arXiv preprint arXiv:2207.13988* (2022).

[12] Matej Ulčar and Marko Robnik-Šikonja. “SloBERTa: Slovene monolingual large pretrained masked language model”. In: *Proceedings of the 24th International Multiconference Information Society - IS 2021, Data Mining and Data Warehouses - SiKDD.* 2021.

[13] Linting Xue et al. “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2021, pp. 483–498.

[14] Aleš Žagar and Marko Robnik-Šikonja. “Slovene SuperGLUE Benchmark: Translation and Evaluation”. In: *Proceedings of the Language Resources and Evaluation Conference.* 2022, pp. 2058–2065.

[15] Matjaž Zupanič et al. *Cross-lingual Question Answering with Transformers.* Assignmnet in the NLP course, University of Ljubljana, Faculty of Computer and Information Science. [20 June 2022]. URL: https://github.com/mtzcorporations/NLP_TeamJodka.

Social Media Analysis for Assessing Resilience

Aljaž Osojnik
aljaz.osojnik@ijs.si

Bernard Ženko
bernard.zenko@ijs.si

Martin Žnidaršič
martin.znidarsic@ijs.si

Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia

ABSTRACT

In this paper, we describe tools developed to investigate the potential use of social media analysis for resilience assessment. We focus on tweets as a data source and apply sentiment analysis, topic detection and filtering approaches. We present computed aggregates with potential information on resilience, and a web application that was made for the use of domain experts. Finally, we discuss preliminary user feedback and lessons learned about the applicability of our approach.

KEYWORDS

social media, sentiment analysis, resilience, disaster management

1 INTRODUCTION

Resilience is generally understood as the ability to adapt and recover after a disruptive event. In this work, we focus on resilience of communities in the context of disaster management, and adopt the following UN definition: *Resilience is the ability of a system, community or society exposed to hazards to resist, absorb, accommodate, adapt to, transform and recover from the effects of a hazard in a timely and efficient manner, including through the preservation and restoration of its essential basic structures and functions through risk management* [7]. Resilience research aims to develop strategies not focused on isolated risks, such as earthquakes or fires, but on approaches that subsume and address all relevant risks, both natural and man-made. The strategies need to identify and account for different human, social, environmental, economic and technological factors that influence behavior of a community when facing a disaster. The goal is that, by adopting these strategies, communities can perform their intended functions in normal and adverse times. A key element of a resilient community is active involvement of local citizens and their active role in a decision making process.

The EU-funded project RESILOC (Resilient Europe and Societies by Innovating Local Communities, https://www.resilocproject.eu/) aims to develop a holistic framework of studies, methods and software tools that can be used to assess the resilience of a community in practice by Local Resilience Teams (LRT).[1] The final goal is to use this framework to identify new strategies for improving the processes of preparedness of local communities against any kind of hazards.

The key element of the RESILOC framework is a methodology for assessing resilience structured along six dimensions: *Governance, Social, Economic, Infrastructure, Disaster Risk Reduction* and *Environmental.* Each dimension is described in terms of its attributes or *indicators,* the values of which are assessed with *proxies*, which are empirically measurable quantities. Indicators and proxies need to have associated scales and aggregation functions that allow their calculation within the RESILOC tool [5]. For example, the *social dimension* of the resilience of a community could be described with indicators such as *Community engagement, Social connectedness, Trust in authority* and *Risk awareness.* The *Community engagement* indicator could then be assessed with proxies such as *% of population who vote in local elections, Number of NGOs for pre- and post-disaster response per capita* and *% of population undertaking voluntary work.* Notably, the resilience assessment of different communities may include different indicators and proxies. The RESILOC platform seeks to provide an initial set of indicators and proxies, allow for the addition of new ones, and enable their aggregation and visualization.

As some of the indicators mentioned above can also be assessed through proxies based on social media analysis, we developed a tool for investigating this approach. For example, proxies (or their components) assessing the indicator *Trust in authority* could be assessed with techniques such as sentiment analysis. In particular, we could hypothesize that a more positive sentiment in social media posts related to public authorities, e.g., disaster response authorities, is related to higher trust in them (and consequently better adoption of any disaster relief measures they introduce). This is the primary motivation that led the investigation presented in this paper. As our social media data source, we use tweets, posts on the Twitter[2] microblogging platform. These are public, abundant and have well supported APIs for collection and filtering.

The rest of the paper is organized as follows. Section 2 briefly presents the the related work, Section 3 presents the social media data used in our analysis and Section 4 presents the results of said analyses. Section 5 presents our web tool for resilience assessment, which is part of the RESILOC framework. The final section concludes our presentation, summarizes lessons learned during our analysis and provides some avenues for further research.

2 RELATED WORK

Our work relates to two main fields of research. The first one is research on community resilience in the context of disaster management. Parker et al. Parker [14] addresses the problems of measuring and assessing resilience and warns that past attempts to define comprehensive resilience assessment frameworks frequently led to simplifications and focus on a single risk. Particular assessments of resilience can be found in the literature review conducted in the scope of the RESILOC project [11].

[1] An LRT is a team in charge of resilience assessment and risk management of a given community, typically organized within a civil protection organization.

[2] https://twitter.com/

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

The second related field is social media sentiment analysis and associated techniques. Sentiment analysis [9] is a machine learning field, that has benefited from the current rapid development of natural language processing techniques [6] based on large corpora and deep learning developed in the recent years. Sentiment analysis of social media posts has previously been applied to resilience adjacent domains, such as disaster response and management [12, 1].

Research in the cross-section of both fields, i.e., resilience and social media, has mostly focused on investigations on how social media affects community and self resilience [8], including recent examples during the COVID-19 pandemic [16]. Using social media analysis to assess resilience is, to the best of our knowledge, novel, and we were not able to find any similar tools to the one presented in this paper.

3 DATA

The data used in our analyses and visualizations are tweets that specifically mention target communities (using a predefined keyword), which were collected through the Twitter API for Academic Research[3]. This allowed us to collect all tweets of interest, including those from the past.

Notably, a tweet is not only the posted text, but rather meta-data-rich data object containing a pleathora of information, such as its language, geolocation, author code, etc., and relations to other tweets, e.g., if it is a response to another tweet or a retweet. For our purposes, however, we only gather the unique tweet code/id (field *id*), creation time (field *created_at*), language (field *lang*) and text (field *text*).

The selected tweets are collected in dataset that we use for our analysis. The dataset is recreated during each repetition of the analysis, due to potential tweet removal according to Twitter's privacy mechanisms. Hence, the results shown in the online app are not static but can change with renewed analysis.

In RESILOC, four communities are studied as use-cases: Gorizia (Italy), Catania (Italy), West Achaia (Greece) and Tetovo (Bulgaria). These four communities vary widely in size, which is reflected in the amount of tweets in which they are mentioned. As the latter two communities are mentioned only in a couple of tweets per month, the social media analysis was executed only for the first two. We considered extending the pool of tweets by including tweets that are geo-tagged to the selected communities, but only a small fraction of tweets contain such information.[4]

The data gathering process consists of collection and filtering. In particular, we collect all tweets that contain the name of the community (Gorizia or Catania) in the text of the post during a given time period. These tweets are then filtered based on the language (*lang=it*) to gather local posts and to filter out some of the noise, e.g., caused by posts mentioning people (particularly celebrities) with names that match the two communities.

4 METHODS

There are four main data analysis results the online app: (1) volume and sentiment, (2) frequent tokens, (3) data for specific topics and (4) sentiment aggregates and trends.

Volume is the amount of tweets in a given time period, yearly or monthly, while *sentiment* refers to the yearly or monthly positive, neutral or negative sentiment detected in the tweets with the approach described below, in Section 4.1. *Frequent tokens* are commonly appearing words, numbers or emojis, that we identify on a monthly basis according to the procedure in Section 4.2. The volume and sentiment aggregates are also calculated for specific subsets of tweets called *topics*, which can be either automatically inferred by the mechanism in Section 4.3, or provided by the users.

4.1 Sentiment Analysis

The goal of the sentiment analysis is to assess the sentiment of tweets in a given community, its trends, and variations in sentiment in general and in specific resilience-related topics.

To assess the sentiment of tweets we use two machine-learned classifiers. The first is a three-class classifier, denoted as LOGREG, that classifies tweets as positive, neutral, or negative, and employs logistic regression. It is trained on high-dimensional vector representations of Italian tweets that include weighted words, pairs of consecutive words, 4-character sequences and emoji characteristics as representation elements, as presented in [10]. The second classifier is the two class (positive, negative) FEEL-IT sentiment classifier [2][5] for Italian that uses word or subword series of character representation in a high dimensional vector space. It is a fine-tuned BERT-based model [6] for Italian.

4.2 Frequent Tokens

In addition to the information on volume and sentiment, we also provide the most frequent tokens (words, numbers or emojis) that appear during any given month. These are provided separately for subsets of tweets, which are classified as positive, negative and neutral (when available).

Frequent tokens relate to the concepts mentioned in the tweets and could, as such, be used do discover aspects of resilience relevant to the community, thus potentially serving as proxy candidates. Frequent tokens are computed using the following procedure: (1) the text of all tweets is whitespace split into tokens and cast to lower case, (2) unwanted tokens are removed, (3) tokens are sorted by frequency of occurrence, and (4) the most frequent tokens are selected for presentation.

Unwanted tokens, mentioned in the second step, are the Italian stop-words. These are filtered using the *nltk* library [3], as well as using a custom unwanted tokens list, such as punctuation characters and various versions of the names of the communities. In the final step, we check that the frequent tokens appear in enough different tweets. Namely, repeated tokens in a single tweet all count towards token frequency, but count only once in terms of tweet appearance. We prevent presenting frequent tokens that do not appear in enough individual tweets, with a occurrence check using a predefined threshold.

4.3 Topic Modeling

To identify potential resilience-related topics, we wanted to automatically model topics in the collected tweets. Our motivation was that, given such topics, resilience experts could analyze the corresponding tweets and extract information useful for resilience assessment and proxy construction.

Topic detection or modeling [13] is a common task in natural language processing and aims at discovering topics, e.g., politics, sports, cycling, etc., that appear in a set of text documents, such as news articles or tweets. The assumption is that if a document discusses a specific topic, some words will appear more frequently.

[3] https://developer.twitter.com/en/products/twitter-api/academic-research

[4] In particular, out of 364531 tweets that mention Catania in 2020, only 8777 (approximately 2.4%) were labeled with geo-location meta-data.

[5] Available at https://github.com/MilaNLProc/feel-it.

Topic modeling is an unsupervised classification method, similar to soft or fuzzy clustering, since a document can belong to more than one topic or cluster. One popular algorithm for topic modeling is Latent Dirichlet Allocation (LDA) [15, 4], which takes the number of topics as an input parameter.

After standard preprocessing of tweets required for topic modeling (upper to lower case, removal of URLs, tokenization into words, removal of stopwords and other irrelevant words), we applied the LDA algorithm with a preset number[6] of topics ranging from 3 to 15. We visualized the resulting topics as word clouds and manually inspected them. We sought to identify topics related to resilience, such as sets of tweets discussing actions of authorities in response to natural disasters (floods, fires, etc.) or citizens perception of authorities' ability to act in case of such disasters. Our inspection was inherently subjective and mostly focused on the top-ranked words, although we also considered standard measures for topic evaluation (perplexity and coherence).

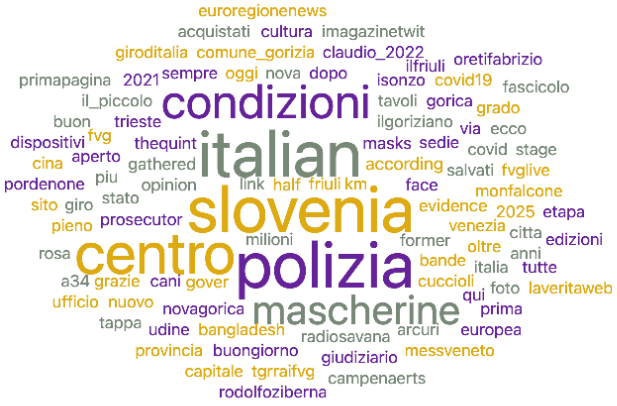


Figure 1: An example of a detected topic. It focuses on COVID-19 measures and conditions for crossing the border between Italy and Slovenia.

The most interesting topic is presented in Figure 1 and includes tweets discussing the COVID-19 measures (*mascherine* is Italian for masks) and conditions (*condizioni*) for crossing the border between Italy and Slovenia, as Gorizia is a border town. Unfortunately, we did not find any topics directly related to resilience, which could be used by resilience experts to assess resilience.

This analysis seems to infer that automatic topic modeling from tweets is likely not very useful for assessing resilience, at least not in the context that we tried to use it, though this might be due to the nature of topic modeling. Namely, the topics that we get obtained were *general*, in that they cover general, rather than resilience specific, concepts, and *unpredictable* in a sense that in some circumstances (and locations) we might obtain resilience related topics and in others not. Ultimately, the automatically constructed topics were not very informative, and, as such, we used topics constructed manually by resilience experts from the involved communities.

4.4 Aggregates and trends

To support facilitate the use of the results of our analyses in the RESILOC platform, several aggregates are explicitly calculated. These aggregates seek to capture the overall sentiment and sentiment trends over various time periods and are available to LRTs, who can use them as inputs for proxy construction. The aggregates are calculated monthly and yearly. The list of aggregates and their descriptions is as follows.

Positive ratio. Ratio of positive tweets vs the total number of total tweets in the current time period.

Neutral ratio. Ratio of neutral tweets vs the total number of total tweets in the current time period.

Negative ratio. Ratio of neutral tweets vs the total number of total tweets in the current time period.

Relative change of volume. Ratio of the number of tweets in the current time period and previous time period.

Relative change of positive tweets. Ratio of the number of positive tweets in the current and previous time period.

Relative change of neutral tweets. Ratio of the number of neutral tweets in the current and previous time period.

Relative change of negative tweets. Ratio of the number of negative tweets in the current and previous time period.

Absolute change in volume. Difference of the number of tweets in the current and previous time period.

Absolute change in positive tweets. Difference of the number of positive tweets in the current and previous time period.

Absolute change in neutral tweets. Difference of the number of neutral tweets in the current and previous time period.

Absolute change in negative tweets. Difference of the number of negative tweets in the current and previous time period.

5 WEB APPLICATION

We make the collected summaries of volume and sentiment analyses, frequent tokens and topic data available through a simple web interface. To access the tool, which is intended for internal use, a user provides a security access token, which determines which community data the user is privileged to view.

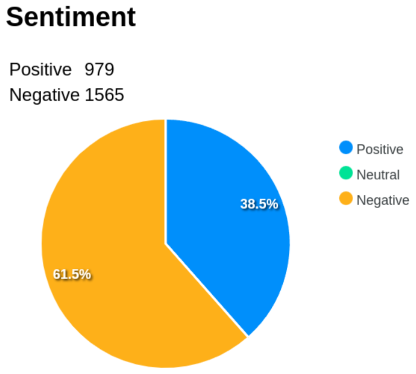


Figure 2: An example of a sentiment distribution as displayed in the web application.

All community data is split into sections based on available classifiers and time intervals. The application provides two options for the time interval, i.e., monthly and yearly views.

The monthly view is composed of the following sections: a total tweet count, tweet count by sentiment accompanied with a corresponding pie chart (as seen in Figure 2), a table of frequent tokens per classified sentiment (as seen in Table 1), a table with the calculated aggregates and trends and, finally, a topics section, that shows values for particular topics. In the final section, each defined topic has a subsection, where the user can see which tokens define the topic, its sentiment distribution and a corresponding pie chart, as shown in Figure 3.

[6] We selected 15 topics as an amount that can be inspected manually.

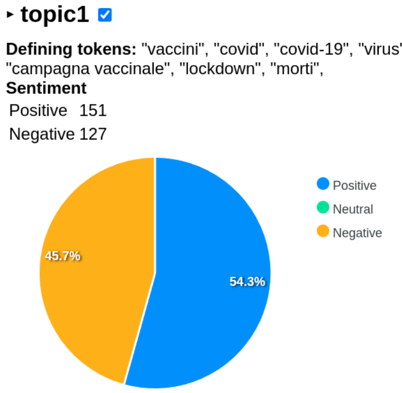
Table 1: An example set of detected frequent tokens.

(a) Positive

Token	Occurence
edizioni	174
europea	141
oggi	139
nova	128
capitale	119

(b) Negative

Token	Occurence
stato	102
recarsi	77
solo	57
x	47
ospedali	39



▸ topic1 ☑

Defining tokens: "vaccini", "covid", "covid-19", "virus", "asugi", "campagna vaccinale", "lockdown", "morti",

Sentiment

Positive 151

Negative 127

Figure 3: An example of a COVID-19 related topic.

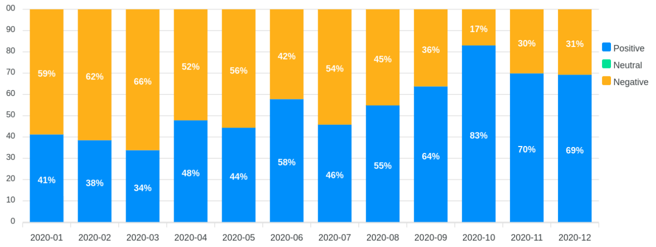


Figure 4: An example of a yearly view of relative sentiment.

The yearly view is composed of the same sections as the monthly view, however, it concerns data for the entire year. In addition to the global pie charts (for sentiment distribution), graphs for the progression of absolute and relative sentiment are shown based on month by month data, as shown in Figure 4.

6 CONCLUSION

We propose a novel approach to resilience assessment based on social media datasets. The analyses and tools described in the paper were developed and presented to potential users in preliminary try-out sessions, i.e., as sprints in Agile software development, as well as discussed with the project consortium's domain experts. The approach is currently being evaluated in the project trials, to quantify its usefulness based on expert feedback.

While automatic topic modeling resulted in some meaningful topics, these were mostly general and very few of them were related to resilience. The users expressed preference for more focused topics, which can now be defined manually, and find the new results interesting and potentially relevant. Interestingly, there is a general preference to not directly use the automatically calculated aggregates of volume and sentiment as inputs to the resilience assessment models, but to be considered and prepared for use by the users, i.e., employed with human oversight.

Analyses such as the ones presented in this paper are only useful for large enough communities that get mentioned in tweets frequently. Furthermore, tweets often do not represent the opinion of the population at large. While they are suitable for analysis, even in real time, their representativity is an issue that needs to be considered when using such data.

ACKNOWLEDGMENTS

This work has been supported by the RESILOC, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 833671. We acknowledge also the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103).

REFERENCES

[1] Ghazaleh Beigi, Xia Hu, Ross Maciejewski, and Huan Liu. 2016. An overview of sentiment analysis in social media and its applications in disaster relief. *Sentiment analysis and ontology engineering*, 313–340.

[2] Federico Bianchi, Debora Nozza, and Dirk Hovy. 2021. "FEEL-IT: Emotion and Sentiment Classification for the Italian Language". In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics.

[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, (Mar. 2003), 993–1022.

[5] Uberto Delprato, Joe Cullen, Thomas Spielhofer, Daniele Del Bianco, Rajendra Akerkar, Nadia Miteva, and Zlatka Gospodinova. 2022. RESILOC Project Deliverable 3.1 – RESILOC Resilience Indicators. Tech. rep. The RESILOC Consortium. Retrieved Aug. 24, 2022 from https://www.resilocproject.eu/wp-content/uploads/2022/05/RESILOC_D3.1_V7.0.pdf.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 4171–4186. DOI: 10.18653/v1/N19-1423.

[7] United Nations Office for Disaster Risk Reduction. 2012. Terminology. Retrieved Aug. 24, 2022 from https://www.undrr.org/terminology/resilience.

[8] Manon Jurgens and Ira Helsloot. 2018. The effect of social media on the dynamics of (self) resilience during disasters: a literature review. *Journal of Contingencies and Crisis Management*, 26, 1, 79–88. DOI: 10.1111/1468-5973.12212.

[9] Bing Liu. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

[10] Matej Martinc, Iza Skrjanec, Katja Zupan, and Senja Pollak. 2017. Pan 2017: author profiling-gender and language variety prediction. In *CLEF (Working Notes)*.

[11] Sjirk Meijer, Jon Hall, Rut Erdelyiova, Marcello Sabanes, Abby Onencan, and Kerstin Junge. 2022. RESILOC Project Deliverable 2.6 – Analysis of different approaches to resilience also outside EU, Section 6. Tech. rep. The RESILOC Consortium. Retrieved Aug. 24, 2022 from https://www.resilocproject.eu/wp-content/uploads/2021/04/RESILOC_D2.6-v6.0_Final.pdf.

[12] Ahmed Nagy and Jeannie A Stamberger. 2012. Crowd sentiment detection during disasters and crises. In *ISCRAM*.

[13] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. Latent semantic indexing: a probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS '98). Association for Computing Machinery, Seattle, Washington, USA, 159–168. DOI: 10.1145/275487.275505.

[14] Dennis J Parker. 2020. Disaster resilience–a challenged science. *Environmental Hazards*, 19, 1, 1–9.

[15] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. 2000. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155, 2, (June 2000), 945–959. DOI: 10.1093/genetics/155.2.945.

[16] Lola Xie, Juliet Pinto, and Bu Zhong. 2022. Building community resilience on social media to help recover from the covid-19 pandemic. *Computers in Human Behavior*, 134, 107294. DOI: 10.1016/j.chb.2022.107294.

Urban Mobility Policy Proposal Using Machine-Learning Techniques

Miljana Shulajkovska
miljana.sulajkovska@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Maj Smerkol
maj.smerkol@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Matjaž Gams
matjaz.gams@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

The world's demography is constantly increasing and most people move to big cities. This growth in urbanization affects people's daily activities by encountering congestion, air and noise pollution, water and energy usage etc. To deal with such issues, city authorities are taking various actions in order to provide the most optimal solution. In that terms testing, evaluating and implementing different scenarios are of great importance that cost money and time at the same time. Therefore, in the era of artificial intelligence, different approaches can be used to automate this process. In this paper, we propose a system that has the potential to automatically propose mobility policies based on previously defined city changes. The decision-makers input the required city changes, while the system outputs a mobility policy that satisfies that specification. To implement the idea, machine-learning algorithms are trained on data produced by a microscopic traffic simulator. The system is tested on data representing the city of Bilbao, where the policies are related to closing the Moyua square in the city centre at a specific time and for different duration, while the city changes are related to air pollution and usage of different means of transport.

KEYWORDS

traffic simulation, artificial intelligence, mobility policy

1 INTRODUCTION

According United Nations report [7] by 2050 two in three people will live in urban areas. However, as cities continue to grow they may face many challenges that affect the daily mobility services and people's movement in general. Therefore, finding a solution that satisfies people's needs is a crucial step toward building a more sustainable mobility system. Currently, many traditional approaches exist that rely on experts' knowledge using results from microscopic simulations. Those approaches include simulation of different mobility policies which are then analysed by the decision-makers. The key issue is how to choose the most appropriate mobility policy that will satisfy the requirements defined by the decision-makers that meet the user's needs. Achieving this goal is impossible without using the help of modern technologies such as machine learning.

In this paper, we propose a system that makes usage of machine learning methods to automate the process of mobility policy suggestions. As decision-makers are interested in achieving a particular set of goals (KPIs) such as reducing CO_2 emissions or increasing the usage of public transport, the system outputs the most appropriate policy that satisfies those requirements.

The rest of the paper is organised as follows. First, an overview of the proposed system is given. Then Section 3 describes the process of collecting the data by giving a brief overview of the simulation tool used, the implemented scenarios and the key performance indicators (KPIs). In Section 4 the machine learning approach is discussed followed by a description of the applied methods and the experimental results. Finally in Section 6 a conclusion is given.

2 SYSTEM

The key concept of the proposed system is to collect data from the microscopic traffic simulator. All the components are shown in Figure 1. The microscopic traffic simulator emulates the behaviour of all the people interacting on the mobility infrastructure. To run the simulator several input files are required such as network and travel demand. Then the output of the simulator is used to calculate the KPIs and later both are used to train the ML model as part of the ML module. The ML module takes as input a required city change defined by the decision-makers and using the trained model outputs a mobility policy that satisfies those requirements.

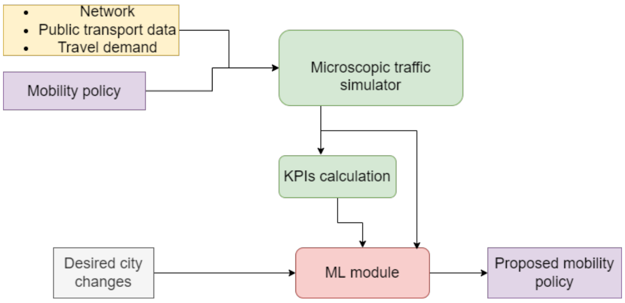


Figure 1: System description

3 DATA COLLECTION

Building a ML module/model that predicts the most suitable mobility policy requires a sufficient amount of data related to a particular set of policy actions and their consequences. In that terms, the main source of data for the proposed system comes from a microscopic traffic simulator as a very common research method for testing and evaluating different mobility situations. In the following sections, a brief overview of the simulation tool is given, and then different scenarios representing specific policies are discussed. Finally, the KPIs used for this study are presented.

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

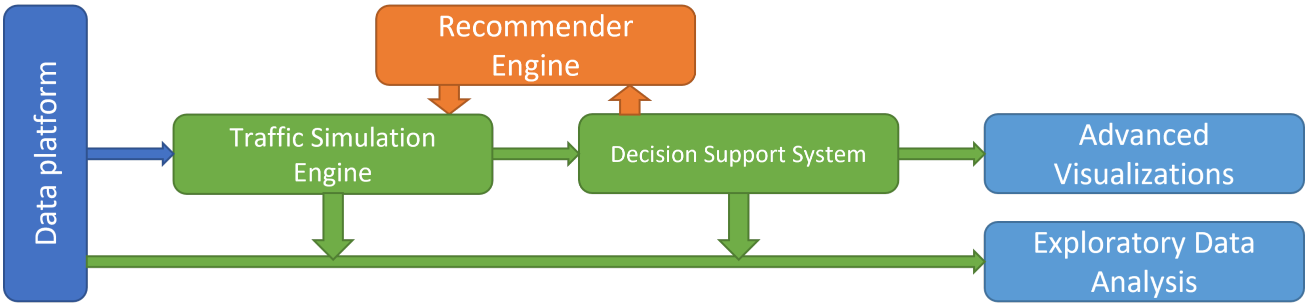


Figure 2: System for Urban mobility policy design architecture.

3.1 Simulation

In this study, MATSim was used as the most suitable microscopic simulation tool [3]. The main concept behind this tool is shown in Figure 3. It consists of an iterative process where plans from the travel demand are executed simultaneously in the mobsim and scored in the scoring module. The score is a metric to evaluate whether a plan is good or bad. It takes into account travelling and waiting time, activity duration etc. More about scoring can be read here [5]. Then a certain number of plans are chosen and modified in the replanning step. After a sufficient number of iterations (in our case 200) an equilibrium is reached where no more plans are evolving, producing higher scores.

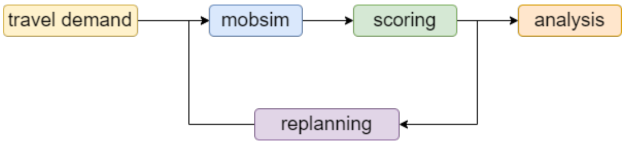


Figure 3: MATSim cycle

A crucial step before running the simulator is to provide data representative of the study area of our interest. The input data is related to the city network map, public transit schedules, travel demand etc. The most challenging part is to construct the travel demand as demographic and other people's movement data is hard to find. Therefore different techniques exist to solve the issue. One approach is to replicate a real population (construct synthetic population) using sample data and marginal distribution, and then assign activity location using origin-destination (OD) matrices. The iterative proportional fitting (IPF) [1] algorithm is used to construct the synthetic population using sample data from EUSILC [2] and demographic data provided by the city. The IPF algorithm is one of the most widely used algorithms for synthetic population reconstruction that combines both datasets (sample data and marginal distributions) to produce weights that show the number of replication of a specific person from the sample data to a specific geographical zone while maintaining the marginal distributions.

The simulation outputs a large amount of data that describes people's movements in form of events. Each event has a type such as "vehicle enters traffic", "vehicle enters/leave link", "person starts activity" etc., time and person's id that performs it. Other output files also exist as histograms of travel/wait times, usage of different transport modes etc. All this data is used in the calculation of KPIs for the ML module.

3.2 Scenarios

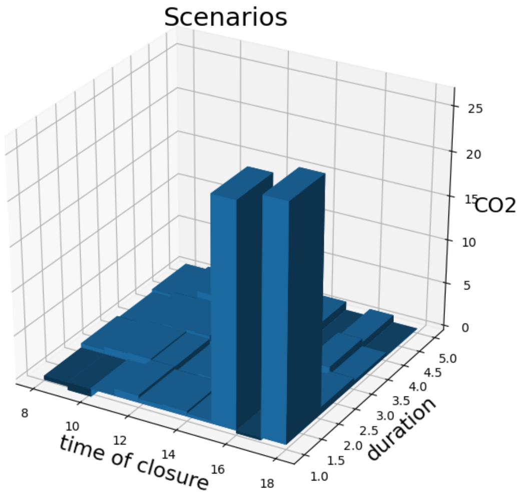


Figure 4: Scenarios CO_2

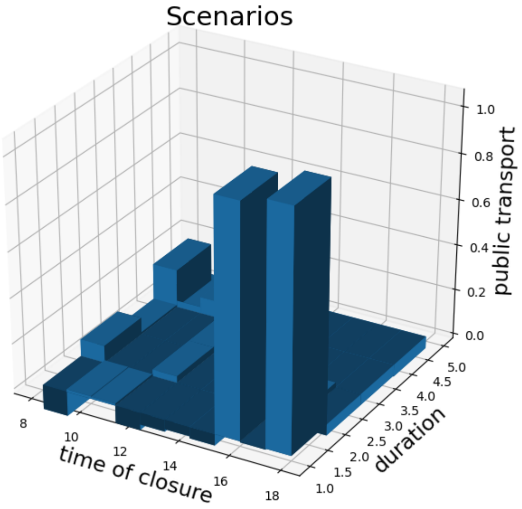


Figure 5: Scenarios Public Transport

We have tested 40 different situations of one policy represented by the closure of the Moyua square from 8 am to 5 pm with a different duration from one to four hours. Results of the applied policy are shown in Figure 4 and Figure 5.

On both plots, the x-axis represents the start time of closure, while the y-axis represents the duration of closing the square for transport in hours. On the first bar plot, the z-axis shows the changes in percents of CO_2 warm emissions and the z-axis on the second bar plot shows the changes in percents in public transport usage compared to a situation when no changes in the city are applied. In terms of reducing the CO_2 emissions, the best scenario is if we close the square at 9 am for 1 hour. In that case the CO_2 has lowered for 0.9%. On the other hand, if the square is closed from 5 pm to 6 pm for traffic, the city gets congested in the surrounding areas and the CO_2 warm exhausts are increased by 26% even though the public transport usage has increased and car usage has decreased as shown on the second bar plot.

3.3 KPIs Calculation

The key performance indicators (KPIs) represent the objectives defined by the decision makers that need to be achieved. In collaboration with the city, a set of KPIs was defined as follows:

- **Air pollution**: CO_2, NOx, PM cold/warm emissions.
- **Usage of different modes**: car, bicycle, public transport, walk.
- **Bike safety, bikeability**

The first set of KPIs related to air pollution is modelled using MATSIm additional emission package [6]. The emissions are calculated using HBEFA (Handbook on Emission Factors for Road Transport) database [4] in combination with the simulation output. As air pollution is caused by different contributions of road traffic the emissions module considers both warm and cold emissions. Warm emissions are emitted while driving and depend on driving speed, stop duration, and vehicle characteristics while cold emissions are emitted during the warm-up phase and are dependent on the engine's temperature.

The second set of KPIs related to the usage of different modes of transport is produced during the simulation, while bike safety and bikeability are calculated from the simulation results.

4 MACHINE-LEARNING FOR POLICY PROPOSAL

The developed system proposes a ML module that helps decision-makers in suggesting mobility policies that satisfy a set of predefined KPIs. As mentioned before, the main source to train the ML models comes from the microscopic traffic simulator. The data used to train the models is shown in Table 1. The KPIs represent the features, while the policies, i.e., scenarios are treated as target variables where the start time and duration of closure are discretized into 30 and 15 minutes intervals respectively. By doing so, the most suitable scenario will be predicted according to the pre-selected KPIs values.

Table 1: Features and target variables

Features		Target variables
CO_2 warm	CO_2 cold	start time of closure
NOx warm	NOx cold	duration of closure
PM warm	PM	
car trips	bike trips	
PT trips	walk	
bikeability	bike safety	

4.1 Methods and Results

Since the target variables are continuous and there are multiple of them, we deal with a multi-output regression problem. This involves predicting more numerical values at the same time which limits the usage of many algorithms that are designed in predicting only a single numerical value. However, to solve the problem several solutions exist:

- Multi-output regression algorithms
- Wrapper multi-output regression algorithms
  - Direct multi-output regression
  - Chained multi-output regression

One approach is to use regression algorithms that support multiple outputs directly such as linear regression, k-nearest neighbours, decision tree, random forest etc. The other approach is to divide the multi-output regression problem into multiple subproblems (wrapper multi-output regression) and then deal with single regression problems. On one hand, there is direct multi-output regression where independent models are developed for the prediction of each numerical value. On the other hand, the chained multi-output regression consists of dependent models when predicting each numerical value.

To evaluate the models a cross-validation technique was used. The mean absolute error (MAE) performance metric is used as a score. The mean and standard deviation of the MAE are calculated across all the folds and all repeats. Table 2 shows the results of inherently multi-output regression algorithms. K-Nearest Neighbours proved the best results with a mean value of MAE of 3.743 and a standard deviation (SD) of 0.852, which means that the difference between the predicted and true value is approximately 4-time intervals or 2 hours for start hour and 1 hour for the duration.

The plot on Figure 6 depicts the difference between true (blue dots) and predicted (red dots) values for 30% of the data. The x-axis represents the start hour, while the y-axis represents the duration of the closure. The smallest error (best-predicted instance) between a true and predicted value is marked with black and yellow dot respectively. Table 3 and Table 4 show results for direct and chained multi-output regression respectively where random forest and k-nearest neighbours proved the best results in both cases.

By applying different ML models, we examined which algorithm can prove the best results in predicting a mobility policy that satisfies a set of predefined city changes. On the other hand, when discussing the simulation results in Section 3.2 we concluded that closing the Moyua square in the afternoon (from 4 pm to 7 pm) decreases the CO_2 for 1%. Also if the square is closed at 11 am, the car usage decreases by 0.12% reducing the CO_2 emissions by 0.3%. Both situations can be implemented to achieve a more sustainable city but which one is better depends on the goals that the city wants to accomplish. If the aim is to decrease the CO_2 in the afternoon peak hours, the first situation is better. The latter situation provides better results if the city is interested in reducing car usage and increasing the usage of public transport. Therefore, selecting the best option when having multiple criteria that need to be satisfied is hard when only human knowledge is included. Additionally, the number of scenarios included in this work is limited as not every possible situation can be tested due to computational and time costs.

Table 2: Inherently Multi-Output Regression Algorithms

Model	MAE (mean)	MAE (SD)
Linear Regression	4.472	2.565
K-Nearest Neighbours	3.743	0.852
Decision Tree Regression	4.367	1.083

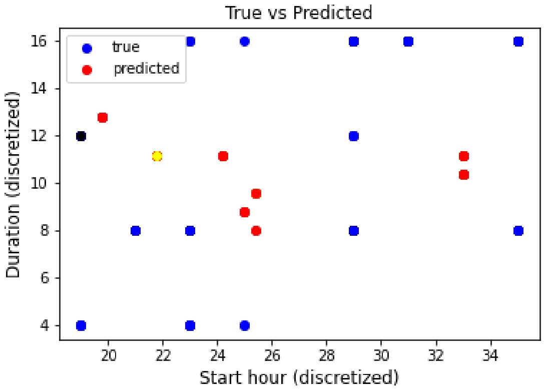


Figure 6: True vs Predicted Data

Table 3: Direct Multi-Output Regression

Model	MAE (mean)	MAE (SD)
Linear Support Vector Regression	5.684	4.709
Random Forest Regression	3.590	0.967
Linear Regression	4.472	2.565
K-Nearest Neighbors Regression	3.743	0.852
Decision Tree Regression	4.333	1.274

Table 4: Chained Multi-Output Regression

Model	MAE (mean)	MAE (SD)
Linear Support Vector Regression	5.718	4.364
Random Forest Regression	3.607	0.931
Linear Regression	4.472	2.565
K-Nearest Neighbors Regression	3.690	0.829
Decision Tree Regression	4.325	1.118

5 CONCLUSION

In this paper, we presented an approach for proposing mobility policies in an automatic way. First, an overview of the system was given. Then the simulation tool was described. 40 variations of one policy for closing the Moyua square in the centre of Bilbao for transport were simulated and evaluated. The variations refer to the start hour and duration of the closure. On the simulation data, the desired KPIs were calculated which together with other simulation output data were used as input to the ML models. Since the ML models output multiple variables (start hour and duration of closure) the problem becomes multi-output regression and limits the usage of many ML algorithms that are developed to deal with a single target. Therefore, two approaches were presented: multi-output and wrapper multi-output regression algorithms. Applying both sets of algorithms, the best results proved random forest regression using the chained method from the second approach.

6 FUTURE WORK

In order to provide the decision-makers with more options when implementing different strategies, in future work different areas around Moyua square will be closed. Closing multiple streets around the square might help in reducing the air pollution in the centre and in the city in general. Also, it can contribute to reducing congestion and other relevant KPIs during peak hours. Therefore, additional target variables such as the city areas to be closed and the length of the streets inside will be added. Since more simulation needs to be executed, three servers will be used to reduce the computational time. By doing this, the dataset will be expanded and more options will be available to the decision-makers in implementing the best scenario that meets the people's needs. Moreover, a mobility expert validation will be included in testing the ML module which will contribute to a more detailed analysis of the ML results.

ACKNOWLEDGMENTS

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 870338. The authors also acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0209).

REFERENCES

[1] Abdoul-Ahad Choupani and Amir Reza Mamdoohi. 2016. Population synthesis using iterative proportional fitting (ipf): a review and future research. *Transportation Research Procedia*, 17, 223–233. International Conference on Transportation Planning and Implementation Methodologies for Developing Countries (12th TPMDC) Selected Proceedings, IIT Bombay, Mumbai, India, 10-12 December 2014. DOI: https://doi.org/10.1016/j.trpro.2016.11.078.

[2] Eurostat. 2020. Eu statistics on income and living conditions microdata 2004-2019, release 2 in 2020. en. (2020). DOI: 10.2907/EUSILC2004-2019V.1.

[3] 2016. *Introducing matsim. The Multi-Agent Transport Simulation MATSim.* Ubiquity Press, (Aug. 2016), 3–8.

[4] Mario Keller, Stefan Hausberger, Claus Matzer, Philipp Wüthrich, and Benedikt Notter. 2017. Hbefa version 3.3. *Background documentation, Berne*, 12.

[5] Kai Nagel, Benjamin Kickhöfer, Andreas Horni, and David Charypar. 2016. A closer look at scoring. (Aug. 2016). DOI: 10.5334/baw.3.

[6] 2016. *Emission modeling. The Multi-Agent Transport Simulation MATSim.* Ubiquity Press, (Aug. 2016), 247–252. DOI: 10.5334/baw.36.

[7] 2017. *Population Facts No. 2017/4, October 2017: The impact of population momentum on future population growth.* https://population.un.org/wpp/Publications/Files/PopFacts_2017-4_Population-Momentum.pdf.

IMF Quality Assurance of Mammograms Using Deep Convolutional Neural Networks and Transfer Learning

Gašper Slapničar
gasper.slapnicar@ijs.si
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Peter Us
peter@u-s.si
UpDev d.o.o.
Kranj, Slovenia

Erna Alukić
erna.alukic@zf.uni-lj.si
Faculty of Health Sciences
University of Ljubljana
Ljubljana, Slovenia

Nejc Mekiš
nejc.mekis@zf.uni-lj.si
Faculty of Health Sciences
University of Ljubljana
Ljubljana, Slovenia

Miha Mlakar
miha@kaiber.si
kAIber d.o.o.
Ljubljana, Slovenia

Janez Žibert
janez.zibert@zf.uni-lj.si
Faculty of Health Sciences
University of Ljubljana
Ljubljana, Slovenia

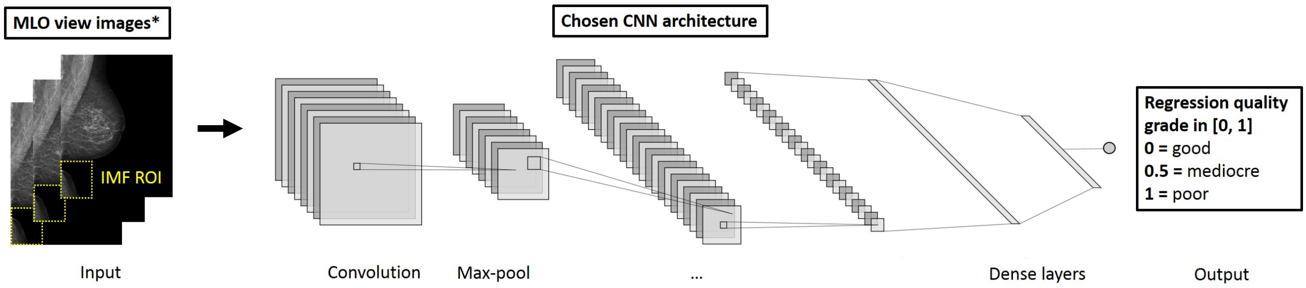


Figure 1: Pipeline of our proposed system. MLO mammograms are taken as input, fed into a chosen CNN architecture, which then outputs a quality grade in an ordinal regression task.

ABSTRACT

Quality assurance (QA) of mammograms is of vital importance, since they are the de-facto method used by doctors for detection of breast cancer and other tissue abnormalities. Despite this, there is a distinct lack of both experts and tools for this task. We thus investigated a deep-learning-based approach using convolutional neural networks for prediction of the inframammary fold (IMF) quality grade, which cannot be measured and determined quantitatively with rules (e.g., if some point x cm from edge, then grade y). We showed in a 5-fold cross-validation experiment that a relatively simple model can achieve respectable performance in terms of root mean squared error (RMSE), area under the ROC curve (AUC) and accuracy, predicting the IMF grade with 3 possible values. Finally we also showed that the model in fact derives features from the relevant ROI also looked at by the experts, hinting at real-world usefulness of such a QA model.

KEYWORDS

mammography, quality assurance, ordinal regression, neural networks, deep learning

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

1 INTRODUCTION

Mammography is the process of using low-energy X-rays to examine human breast tissue for diagnosis and screening, with the typical goal being early detection of breast cancer through detection of anomalies in the tissue [3]. The procedure consists of compression of breast tissue using a dedicated mammography device with the aim of reducing and evening out the tissue thickness that X-rays must penetrate, in turn reducing the required radiation dose. There are two common views in which a mammogram is recorded, namely craniocaudal (CC) and mediolateral oblique (MLO). The former captures the breast tissue in a top-down direction along the pectoral muscle plane, while the latter captures it at an angle sideways.

The importance of regular mammographic screening can not be overstated, as it the de-facto method used by doctors for early breast cancer or other tissue-anomaly detection (e.g., tumors). However, the procedure itself is rather involved and can be tedious for the patient. Subsequently, it is of utmost importance to ensure high quality of taken images, as it is highly undesirable for a patient to have to revisit and repeat the procedure. To this end there are a number of guidelines available that are being followed by the radiologists with the aim of minimizing the amount of low-quality images taken. Some quality metrics can be quantified and measured precisely, while others are more subtle and often left to the expertise of the professionals. One such elusive metric is the Inframammary Fold (IMF). IMF is the inferior border of the breast and the crease between the breast and abdominal tissue.

It serves as an important anatomical landmark on an MLO mammogram to provide assurance to the radiographer that all of the posterior breast tissue has been included.

In Slovenia, only a single radiographer is responsible for weekly grading of randomly sampled mammographic segments from that week, as a part of the DORA oncology program. This is inefficient and the grade itself can be subjective, especially the metrics that are not clearly defined, such as IMF. There is thus a need for automated tools that would help radiologists with quality assurance (QA), optimizing the process while also potentially serving as a training tool for improvements in quality of mammograms being collected.

In this paper we highlight the importance and lack of QA methods for mammograms, and develop two deep convolutional-neural-network (CNN) computer vision models, aiming to recognize and successfully predict the IMF quality metric. The latter is known to be especially tricky and subjective. We evaluate and compare the performance of our proposed models on a custom dataset collected in Slovenia.

The rest of this paper is organized as follows: we first highlight related work about QA in mammography, together with existing computer vision methods that are important for QA in Section 2; in Section 3 we describe our dataset; Section 4 details our methodology and experimental results; and in Section 5 we summarize and discuss the implications with future directions for our work.

2 RELATED WORK

To ensure the key goals of mammography are achieved, quality assurance must be adopted in order for the mammograms to be suitable for diagnosis. In the past decades, several standards have been developed nationally and internationally to this end. A review study by Reis et al. [3] presents an overview of these, showing importance of both technical and clinical aspects, especially with the development of digital mammography.

In terms of QA, there are specific keypoints or region segmentations that are often required for individual grades. One such is the segmentation of pectoral muscle in the MLO view, which was traditionally segmented using pixel thresholding and region growing algorithms. Recently however, with the rise of deep learning, this task was successfully resolved. Soleimani et al. [6] proposed a two-stage algorithm that predicts precise pectoral muscle boundary using a CNN. Evaluating on three datasets they achieved average values of dice similarity and accuracy of 97% and 99% respectively.

Other researchers focused on the final task directly - prediction of anomalies related to cancer. For instance Abel et al. [1] proposed a CNN architecture for detection of abnormal axillary lymph nodes, which is a specific abnormality in the tissue. They reported accuracies of 96% for detection of suspicious lymph nodes. Shen et al. [5] have importantly shown that such end-to-end networks are not only performing well, but can be transferred between different datasets, for instance CBIS-DDSM and INBreast, hinting at good generalization capabilities.

Despite all existing work, we noticed a distinct lack of research dealing with machine learning QA prediction, meaning models that output grades of mammograms rather than try to predict some other subsequent outcome. The quality itself is however vital both for physicians or other models taking mammograms as inputs, as good quality mammograms are useful for detections, while poor quality is not only difficult for diagnosis, but requires a repeat measurement which is "expensive" for everyone involved.

3 DATA DESCRIPTION AND PREPROCESSING

The dataset used in our work was collected in the previously mentioned DORA program between 2011 and 2013 and graded by the single expert using a modified PGMI (Perfect, Good, Moderate, Inadequate) grading system, where there were just three grades (Good, Moderate, Inadequate/Poor). All the mammograms were anonymized and held by the Faculty of Health Sciences. In total there were 4928 mammograms, 2424 in the CC view and 2424 in the MLO view, each view in turn having a left (L) and right (R) image, each with a corresponding ground truth label. Example images can be seen as input in Figure 1.

Initially these images were saved in the widely-used Digital Imaging and Communications in Medicine (DICOM) format, which contains the image itself with a lot of corresponding metadata. These initial images are grayscale and come in varying resolutions, which is problematic for CNN inputs, which are expected to be of a constant shape. Subsequently we first extracted the images and standardized the resolution to 256 x 256 pixels, resaving them in lossless .png format. Some information loss can not be avoided however, since we are substantially decreasing resolution in this step.

Each image is by default also equipped with letters in the top-right or top-left corner of the image itself, denoting the view and side (e.g., CC-L or MLO-R). Since our computer vision models might use this information to learn some data-specific pattern between the view and quality, we semi-manually removed these letters by zeroing pixels in an empirically determined region, leaving only the breast tissue in each image.

Finally, some of the images have important keypoints or areas that are very difficult to see, especially with the naked eye. We thus investigated Adaptive Histogram Equalization methods, more specifically Contrast-Limited AHE (CLAHE), which was reported in related work to help substantially with visibility of important regions in mammograms [2]. The effects can be seen in Figure 2.

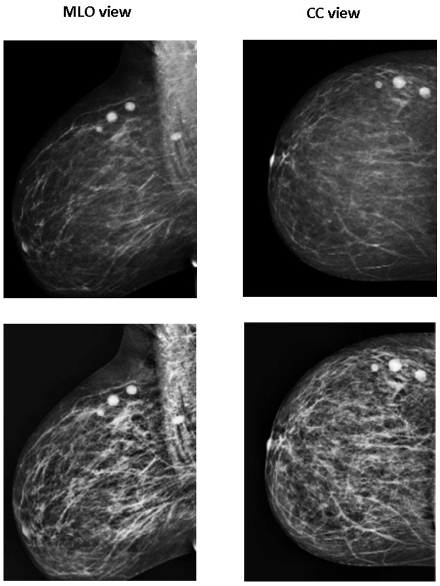


Figure 2: Effects of CLAHE image preprocessing on mammograms in both views. Top row are originals, bottom rows are preprocessed.

For our class label we used the one given by the expert for the IMF. It is a numeric value from 1 to 3, where 1 = good, 2 = moderate and 3 = poor. It is problematic since it is known to be subjective, but still the best we could obtain. It is also quite difficult to visualize clearly for a non-expert, but we show the relevant region of interest (ROI) looked at by the experts in Figure 1. Ideally the ground truth grade would be a voted value obtained by several experts, but that was not feasible. Since IMF is only evaluated on the MLO view, we used just those mammograms in further analysis.

4 METHODOLOGY AND RESULTS

The learning problem itself seems like a typical classification, however, after giving it some thought, we realized it is better to set it up as an ordinal regression problem, as we are predicting a range of grades. We thus mapped our class label values from discrete 1, 2 and 3 to the [0.0, 1.0] interval, where grade 1 = 0.0, grade 2 = 0.5 and grade 3 = 1.0. We thus obtain a numeric value from the network which gives us not only the class information but also the distance between prediction and ground truth (e.g., prediction of 0.96 is much better compared to prediction of 0.76, given ground truth 1.0).

Once our data was finalized in terms of inputs and outputs, we focused our attention towards a model. Related work dealing with visual tasks in general, as well as with mammograms specifically, shows convincing dominance of CNNs in the past decade. Subsequently we decided to investigate such architectures.

Our aim was to start with a simple architecture consisting of three 2D Conv layers with 16, 16 and 8 kernels (each of size 9 x 9), intermediate max pooling layers with kernel size 2 and stride 2, and one fully connected layer with 1000 neurons on top. We used batch normalization and dropout as commonly-used mechanisms to prevent overfitting. ReLU was used as the activation function and the network was trained for 100 epochs.

We then wanted to compare such a simple architecture with a more complex CNN. We decided to attempt a transfer learning approach, where we based it on the known VGG19 model trained on ImageNet. We replaced the final layer with two fully connected layers to instead predict our IMF grade, while keeping the bulk of the model intact with existing weights. Hyperparameters were mostly left at default values, except for learning rate which had a linear decay implemented in our simple architecture.

Initially the full data was split into training (80%), validation (10%) and test sets (10%), as traditional. However, using a random train-validation-test split can be volatile and is undesirable in terms of making any conclusions about robustness of the model. We thus instead used the 5-fold cross validation (CV) evaluation setup, where k-1 groups are taken for training and the remaining group is left for testing in each of the 5 iterations. Additionaly it was ensured that the distribution of labels is kept in each group, meaning we did a stratified split. This was important in our experiment since the class label distribution is relatively skewed (good / 1 / 0.0 = 65%, moderate / 2 / 0.5 = 30%, poor / 3 / 1.0 = 5%) and we are especially interested in the bad examples, which are in vast minority.

For evaluation metrics we used root mean squared error (RMSE) and area under the curve (AUC of ROC curve), while also looking at the classificaion accuracy (transforming from ordinal regression back to classification) as it is the most intuitive metric. In terms of AUC, we always compared all possible pairs of grades, meaning 1v2, 1v3 and 2v3. The most important is to have good separation between poor images and everything else, since those are the most problematic, while moderate images could be close to either good or poor. Numeric results are given in Table 1 and the ROC curves for all three cases (of the better performing model) are given in Figures 3, 4 and 5.

Table 1: Numeric results from the 5-fold CV using our simple model and transfer learning with VGG19.

		Simple model			
Fold Nr.	RMSE	AUC 1v3	AUC 1v2	AUC 2v3	Accuracy
1	0.26	0.90	0.79	0.78	0.93
2	0.25	0.98	0.80	0.92	0.95
3	0.24	0.98	0.79	0.95	0.91
4	0.23	0.97	0.78	0.89	0.96
5	0.24	0.94	0.78	0.84	0.95
Avg.	**0.24**	**0.95**	**0.79**	**0.88**	**0.94**
		Transfer VGG19 model			
1	0.29	0.78	0.73	0.60	0.79
2	0.27	0.89	0.73	0.77	0.75
3	0.28	0.85	0.70	0.71	0.66
4	0.28	0.94	0.65	0.92	0.70
5	0.30	0.85	0.69	0.73	0.77
Avg.	**0.28**	**0.86**	**0.70**	**0.75**	**0.73**

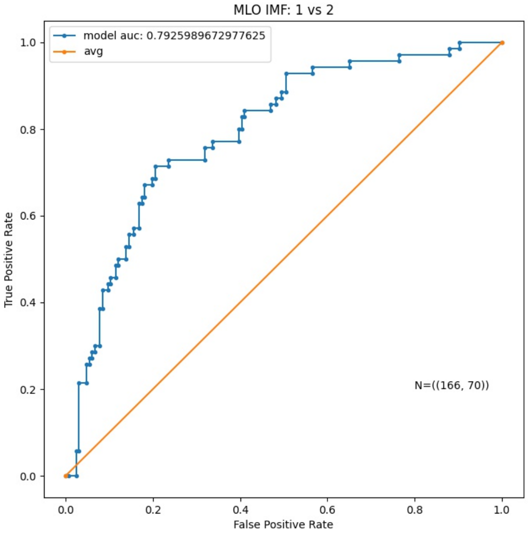


Figure 3: ROC curve of the better performing (simple) model for 1v2 class combination.

5 DISCUSSION AND CONCLUSION

Looking at the results, we can initially observe the overall better performance of our simple model compared to the pre-trained VGG19 transferred to our domain. All the metrics are more stable across fold while also achieving better overall values. Since we were especially interested in the separation of the good and poor mammograms (grade 1 vs. grade 3), we can also see that this

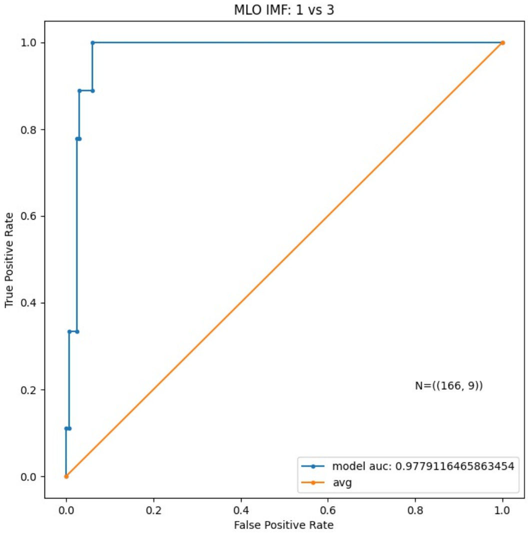


Figure 4: ROC curve of the better performing (simple) model for 1v3 class combination.

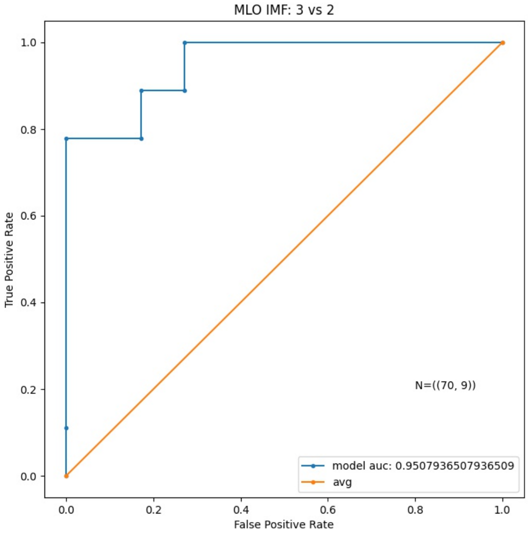


Figure 5: ROC curve of the better performing (simple) model for 2v3 class combination.

separation is indeed consistently successful and seems relatively robust. As expected the separation of moderate (grade 2) mammograms from others is more challenging, but still can be done reasonably well.

Deep learning models are often criticized for being black-box and not offering explainability for their decisions. We also wanted to do a quick investigation of this by using the Grad-CAM approach [4], which is a popular technique for producing “visual explanations” for decisions from a large class of CNN-based models in the form of a heatmap showing where the model focused the most on an image. An example is shown in Figure 6, where we can see the model commonly focused on the relevant IMF ROI that is also focused by the experts. However, this focus was not exclusive, meaning it did not focus just that region and also it wasn’t the same on all images, but still rather consistent, which is a good indicator that the model actually learned the relevant features for IMF QA.

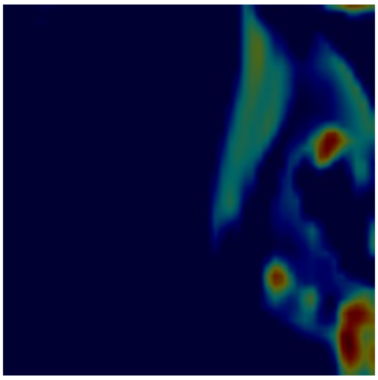


Figure 6: Grad-CAM heatmap showing the areas on the mammogram that were focused by the model to derive features.

To summarize, we investigated the possibility of CNN-based IMF QA for mammograms, looking at a simple CNN model and a transferred slightly-modified VGG19 model. The simple CNN architecture achieved respectable results in terms of several metrics and importantly also focused on the correct part of the image without any guidance, hinting that it learned the relevant features also looked at by the experts. Extensions to prediction of other QA grades might allow for a system that could help with continuous QA, as well as expert training.

ACKNOWLEDGMENTS

We would like to acknowledge the DORA program for providing us with the dataset. Additionally we would like to thank colleagues from the Faculty of Health Sciences for helping with understanding and interpretation of data. Finally a special thanks goes to kAIber d.o.o. and UpDev d.o.o. for participating in the data analysis and modelling.

REFERENCES

[1] Frederik Abel, Anna Landsmann, Patryk Hejduk, Carlotta Ruppert, Karol Borkowski, Alexander Ciritsis, Cristina Rossi, and Andreas Boss. 2022. Detecting abnormal axillary lymph nodes on mammograms using a deep convolutional neural network. *Diagnostics*, 12, 6, 1347.

[2] Etta D Pisano, Shuquan Zong, Bradley M Hemminger, Marla DeLuca, R Eugene Johnston, Keith Muller, M Patricia Braeuning, and Stephen M Pizer. 1998. Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *Journal of Digital imaging*, 11, 4, 193–200.

[3] Cláudia Reis, Ana Pascoal, Taxiarchis Sakellaris, and Manthos Koutalonis. 2013. Quality assurance and quality control in mammography: a review of available guidance worldwide. *Insights into imaging*, 4, 5, 539–553.

[4] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-CAM: visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128, 2, (Oct. 2019), 336–359. DOI: 10.1007/s11263-019-01228-7.

[5] Li Shen, Laurie R Margolies, Joseph H Rothstein, Eugene Fluder, Russell McBride, and Weiva Sieh. 2019. Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9, 1, 1–12.

[6] Hossein Soleimani and Oleg V. Michailovich. 2020. On segmentation of pectoral muscle in digital mammograms by means of deep learning. *IEEE Access*, 8, 204173–204182. DOI: 10.1109/ACCESS.2020.3036662.

Vehicle Axle Distance Detection From Time-series Signals Using Machine Learning

David Susič
david.susic@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Blaž Erzar
blaz.erzar@gmail.com
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Nika Čelan
nika.celan8@gmail.com
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Gašper Leskovec
leskovecg@gmail.com
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Žiga Kolar
ziga.kolar@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

Martin Konečnik
martin.konecnik@cestel.si
Cestel cestni inženiring d.o.o
Špruha 32
Trzin, Slovenia

Domen Prestor
domen.prestor@cestel.si
Cestel cestni inženiring d.o.o
Špruha 32
Trzin, Slovenia

Matjaž Skobir
matjaz.skobir@cestel.si
Cestel cestni inženiring d.o.o
Špruha 32
Trzin, Slovenia

Matjaž Gams
matjaz.gams@ijs.si
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

In this study, we compare a signal decomposition and a convolutional autoencoder approach in determining vehicle axle distances. Our dataset consists of 62076 instances of vehicles crossing a bridge. Each vehicle is detected by eight identical sensors placed under the bridge that record time series vibration data. We compare our results to those computed using an expert's model, which we consider to be the ground truth. The signal decomposition approach achieves accuracies of up to 0.89, 0.98, and 1, with the calculated distances matching the expert model within 2%, 5%, and 10%, respectively. The convolutional autoencoder, on the other hand, achieves accuracies of up to 0.97, 0.99, and 1 with the same error margins compared to the expert model.

KEYWORDS

vehicle detection, axle distance, neural network, peak detection, machine learning

1 INTRODUCTION

In order to accurately weigh the vehicles crossing the bridge and determine if they weigh too much and damage the road, the vehicle speed, the number of vehicle axles, and their in-between distances must first be determined.

In recent years, many types of sensors have been used for vehicle detection. These include acoustic sensors, inductive loop sensors, strain sensors, magnetic sensors, and imaging sensors. Researchers around the world have developed various methods for using sensor data for vehicle axle detection, weight detection, and classification.

The authors of Marszalek et al. [5] measured vehicle axle distances based on multifrequency impedance measurement of a slim inductive loop sensor. Using test vehicles, they were able to confirm that their method can successfully determine the distances. In the work by Chatterjee et al. [1] they used data from sensors on the bridge and a wavelet-based analysis to determine the axle distances. In the work of Khalili et al. [3], piezoelectric elements were used for a system to detect the weight of vehicles in motion. They used the weight-in-motion system to determine both the axle distances and vehicle weights with sufficient accuracy. Rujin et al. [4] developed a deep learning system for vehicle recognition based on strain sensor data. They were able to classify 11 different vehicle types with a very high average precision.

In this work, we use data collected from a single bridge to test two machine learning approaches for vehicle axle distance detection. The first approach is based on signal decomposition and the second approach is based on the convolutiona neural network autoencoder.

We begin in section 2 with a description of the dataset used in this study. In section 3, we explain our approaches and illustrate them with examples. Results are presented and discussed in section 4. The paper concludes with section 5.

2 DATASET

Our dataset consists of sensor data from vehicles crossing the bridge. The sensors are placed under the bridge in the configuration shown in the Figure 1. The sensors are identical and record the vibrations of the crossing vehicles at a sampling rate of $512 Hz$. In this study, only data from vehicles travelling in lane 1 were used (orange sensors 1–8 in Figure 1). The vibration data from the sensors in the first and last columns (1 and 2) are also used to calculate the vehicle speed. The vehicle speed is calculated by superimposing the signals using a cross-correlation method. Our dataset consists of 62076 instances, where each instance contains data for one vehicle. In addition, each instance also contains the axle distances calculated by the expert model and the times at which each vehicle axle crossed the signal.

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

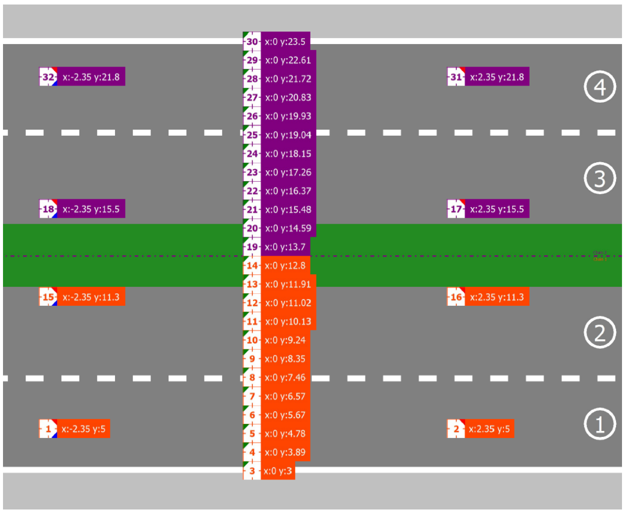


Figure 1: Placement of the sensors on the bridge.

An example of sensor data for a vehicle with 5 axles is shown in Figure 2. The signal peaks correspond to the vehicle axles crossing the sensor, while the amplitude corresponds to the weight of the vehicle on that axle. Although the peaks in the signal correspond to the vehicle axles, the intervening spacing of the peaks generally does not represent the actual axle spacing. Due to interference between the signals from the individual axles, the peaks shift relative to the position of the peaks if the signals from the individual axles were isolated. A small effect of this can be seen in the peak triplet in the right part of the signal in Figure 2. In some cases, where two of the adjacent vehicle axes are very close to each other, the peaks may overlap and become indistinguishable from each other, resulting in a single peak.

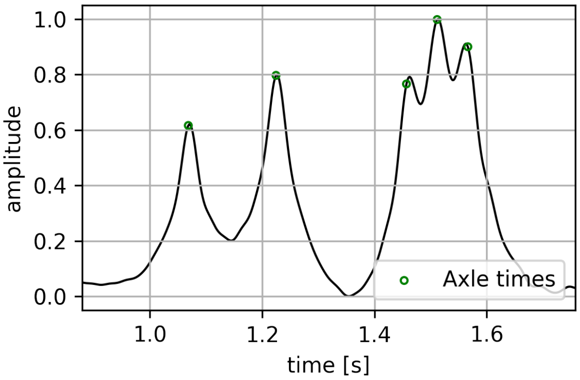


Figure 2: Example sensor data for a vehicle with 5 axles. The green markings correspond to the crossing points of the vehicle axles as calculated by the expert model.

3 METHODOLOGY

The objective of this study is to evaluate the performance of signal decomposition and convolutional autoencoder approaches in computing vehicle axle distances from sensor data. In our two approaches, the signal timing of the axles is first determined and then their intermediate distances are computed based on the vehicle speed.

We compare our results with those of the expert model, which has a measured accuracy of 98% in practise. Our results are considered correct if all calculated axle distances match those of the expert model within the specified margin of error. In addition, we have the option to skip the predictions of the results whose confidence level is below a certain threshold.

3.1 Signal Decomposition

The first step of a signal decomposition approach was to determine the most appropriate signal for each instance. For this purpose, peak detection was performed for all eight signals, calculating the first and second derivatives. A 62076 × 8 matrix was then created, with the eight columns indicating the number of detected peaks from each of the signals. A gradient boosting regression model was then trained on this matrix, the output of which was an array with the correct number of peaks. In addition, eight gradient boosting classifiers were trained, one for each of the signals. The output of each classifier was an array giving the probabilities that the number of detected peaks was correct for that signal.

For each test instance, the regression model first predicted the axis number, rounding the result to the nearest integer. All eight signals were then run, starting with the signal from sensor one. For each signal, a check was made to see if the number of peaks determined matched the number predicted by the regression model. If it did, we checked whether the probability that the number of detected peaks, as predicted by the classifier for that signal, was above a *confidence* threshold. Iteration was stopped if both criteria were met, and the signal was selected as the most appropriate for that instance. This means that in most cases not all signals were checked. Although in principle there could be a signal that would be even more suitable, we found experimentally that in more than 90% of cases signal 1 was the best, followed by signals 2, 5, and 6, in that order. If none of the eight signals met the criteria, the instance was skipped. In our experiments, we used *confidence* values between 0 and 0.997.

After the best signal for each instance was determined, the signals were decomposed into what are called base waves. A base wave is a function designed to have the form of an isolated wave, and can be constructed with three parameters: x-location, scaling in the x-direction, and scaling in the y-direction. The signal decomposition can be defined as an optimization problem where we want to find the best parameters for the base waves. The objective function we want to minimize is the mean square error between the original signal and the sum of the base waves. Once the signal was optimally decomposed, the peaks of the base waves were used to calculate the axle distances. The base wave peaks now correspond to the actual axle times and represent isolated waves, thus their peaks are not shifted by interference. This can also be seen in Figure 3: the green and black vertical lines do not exactly coincide, which is most obvious in the triplet on the right.

Peak detection and models' training was performed using Python 3.7 and libraries Scikit 0.24.2 [7] and Numpy 1.18.5 [2].

3.2 Convolutional Autoencoder

The second approach is a convolutional neural network as an autoencoder. The schematic of the model is shown in Figure 5. The first layer is the input layer. It consists of 4000 nodes, since this was the number of samples of the longest signal, and takes the raw signal as input. The signals with length less than 4000

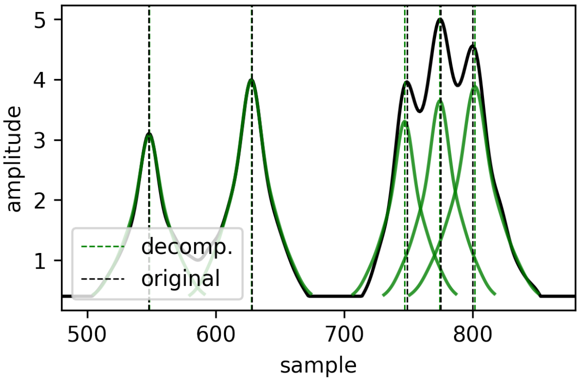


Figure 3: Example of signal decomposition. The black vertical lines represent the peaks as detected during peak detection, while the green lines represent the peaks after decomposition.

were padded with zeros. For all instances, signal 1 was chosen as the input signal, since it worked best in most cases. The encoding part of the autoencoder consists of three convolutional layers with sizes 5, 2 and 3 and the number of filters 8, 16 and 32. Each convolutional layer is followed by a batch normalization and a max-pooling of size 2. The decoder has the opposite structure compared to the encoder and the max-pooling layers are replaced by up-sampling layers. The output layer has the same size as the input layer. The loss function used for model training was a binary cross entropy.

The output for each training instance was a binary array, with ones at the sample locations containing a peak and zeros everywhere else. Thus, for the unseen (test) instances, the model outputs the probabilities for each of the input signal samples to include a peak. An example output for a test instance is shown in Figure 4. It can be seen that the probabilities for the peaks are almost always less than one, but the number of probabilities that are not zero (or not very close to zero) is equal to the number of actual peaks. It can also be seen that the model has learned to shift the peaks where necessary (triplet on the right).

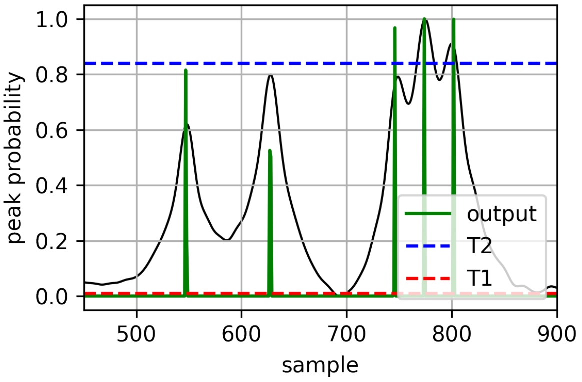


Figure 4: Example output of convolutional autoencoder.

After decoding, we selected lower and upper probability thresholds, *T1* and *T2* (red and blue dashed lines in Figure 4). If all peaks had a probability of at least *T1* and the highest probability was at least *T2*, the axle distances were calculated, otherwise the instance was skipped. In our experiments, *T1* was fixed at 0.01, while we tried values between 0 and 0.7 for *T2*.

Convolutional autoencoding was performed using Python 3.7 and library Tensorflow 2.9.1 [6].

4 RESULTS

Both approaches were tested with 5-fold cross-validation, and the folds were the same in both experiments. The results are shown in tables 1 and 2. They are given for a percentage of skipped instances between 0 and 50 %. The "±x" values in the brackets of the accuracy columns represent percentages within which the calculated axle distances must match those given by the expert models for the prediction to be considered correct. In the Table1, the confidence column corresponds to the *confidence* threshold of the prediction model for the number of peaks, while the T2 column in Table 2 corresponds to the minimum peak probability of the peak with the highest probability. If these criteria are not met, an instance is skipped. The corresponding amounts of skipped instances are given in the skipped columns.

We see that the signal decomposition approach achieves accuracies up to 0.89, 0.98, and 1, with the calculated distances matching the expert model within 2%, 5%, and 10%, respectively. The convolutional autoencoder, on the other hand, achieves accuracies of up to 0.97, 0.99, and 1 compared to the expert model, with the same error margins.

It can also be seen that for both approaches, the accuracies start to converge when about 15% of the instances are skipped, and do not improve significantly even when the percentage of skipping is 50. The convolutional autoencoder generally has higher accuracy than the signal decomposition approach, except in cases where the margin of error is 10 %, in which case the performances of both approaches are similar.

5 CONCLUSION

In this work, we tested a signal decomposition and convolutional autoencoder approach for vehicle axle distances detection using data from eight sensors mounted under the bride. We used a dataset of 62076 vehicles travelling in the same lane. We compared our results to those computed by an expert's model, which we considered to be the ground truth. Using the signal decomposition approach, we achieved accuracies of up to 0.89, 0.98, and 1 for the cases where each vehicle axle distance matched the expert model within 2%, 5%, and 10%, respectively. For the convolutional autoencoder, the accuracies obtained were 0.97, 0.99, and 1 for the same error margins compared to the expert model. The models will be improved in future work to include detection of vehicle axle weights.

ACKNOWLEDGMENTS

This study received funding from company Cestel. The authors acknowledge the funding from the Slovenian Research Agency (ARRS), Grant (PR-10495) and Basic core funding P2-0209.

REFERENCES

[1] Pranesh Chatterjee, Eugene OBrien, Yingyan Li, and Arturo González. 2006. Wavelet domain analysis for identification of vehicle axles from bridge measurements. *Computers & Structures*, 84, 28, 1792–1801. DOI: https://doi.org/10.1016/j.compstruc.2006.04.013.

[2] Charles R. Harris, Jarrod K. Millman, Stefan J. van der Walt, Ralf Gommers, Pauli Virtanen, and David Caurnapeau. 2020. Array programming with numpy. *Nature*, 585, 357–362. DOI: https://doi.org/10.1038/s41586-020-2649-2.

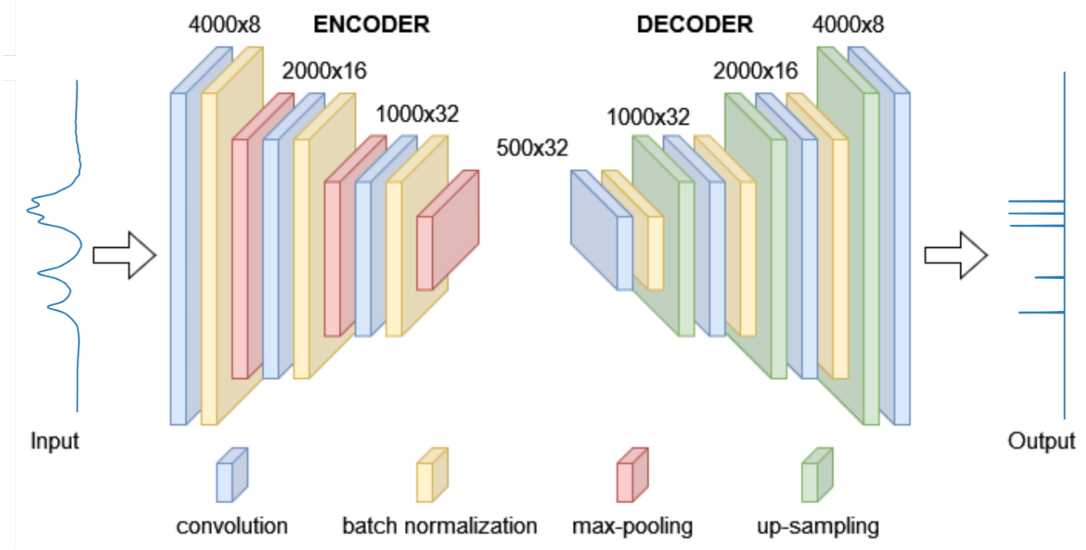


Figure 5: Scheme of the convolutional autoencoder structure.

Table 1: Results for the signal decomposition approach.

Confidence	Skipped	Accuracy (± 2%)	Accuracy (± 5%)	Accuracy (± 10%)
0.0	0.01	0.79	0.88	0.91
0.75	0.05	0.82	0.92	0.95
0.865	0.1	0.84	0.95	0.97
0.915	0.15	0.86	0.96	0.98
0.95	0.2	0.87	0.96	0.99
0.974	0.25	0.88	0.98	0.99
0.977	0.3	0.88	0.98	0.99
0.982	0.35	0.88	0.98	0.99
0.988	0.4	0.88	0.98	0.99
0.996	0.46	0.89	0.98	1.0
0.997	0.5	0.89	0.98	1.0

Table 2: Results for the signal convolutional autoencoder approach.

T2	Skipped	Accuracy (± 2%)	Accuracy (± 5%)	Accuracy (± 10%)
0.0	0.0	0.85	0.88	0.89
0.021	0.06	0.91	0.94	0.95
0.12	0.1	0.94	0.97	0.98
0.431	0.15	0.95	0.98	0.99
0.488	0.2	0.96	0.99	0.99
0.53	0.25	0.96	0.99	0.99
0.566	0.3	0.96	0.99	0.99
0.601	0.35	0.96	0.99	0.99
0.636	0.4	0.97	0.99	1.0
0.665	0.45	0.97	0.99	1.0
0.7	0.5	0.97	0.99	1.0

[3] Mohamadreza Khalili, Gopal Vishwakarma, Sara Ahmed, and Athanassios Thomas Papagiannakis. 2022. Development of a low-power weigh-in-motion system using cylindrical piezoelectric elements. *International Journal of Transportation Science and Technology*, 11, 3, 496–508. DOI: https://doi.org/10.1016/j.ijtst.2021.06.004.

[4] Rujin Ma, Zhen Zhang, Yiqing Dong, and Yue Pan. 2020. Deep learning based vehicle detection and classification methodology using strain sensors under bridge deck. *Sensors*, 20, 18. https://www.mdpi.com/1424-8220/20/18/5051.

[5] Zbigniew Marszalek, Waclaw Gawedzki, and Krzysztof Duda. 2021. A reliable moving vehicle axle-to-axle distance measurement system based on multi-frequency impedance measurement of a slim inductive-loop sensor. *Measurement*, 169, 108525. DOI: https://doi.org/10.1016/j.measurement.2020.108525.

[6] Martín Abadi et al. 2015. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. (2015). https://www.tensorflow.org/.

[7] F. Pedregosa et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

Študija učinkovitosti algoritma za razporejanje terenskega dela

A Study of the Performance of a Fieldwork Scheduling Algorithm

Tea Tušar
Institut "Jožef Stefan" in
Mednarodna podiplomska šola
Jožefa Stefana
Jamova cesta 39
Ljubljana, Slovenija
tea.tusar@ijs.si

Nace Sever
Univerza v Ljubljani
Fakulteta za računalništvo in
informatiko
Večna pot 113
Ljubljana, Slovenija
nace.sever@gmail.com

Aljoša Vodopija
Bogdan Filipič
Institut "Jožef Stefan" in
Mednarodna podiplomska šola
Jožefa Stefana
Jamova cesta 39
Ljubljana, Slovenija
aljosa.vodopija@ijs.si
bogdan.filipic@ijs.si

POVZETEK

Razporejanje terenskega dela je zahteven optimizacijski problem. Za njegovo reševanje smo razvili trinivojski algoritem. Na prvem nivoju evolucijski algoritem razporedi naloge po delavcih, na drugem nivoju hevristika za vsakega delavca razporedi naloge po dnevih, na tretjem nivoju pa algoritem razveji in omeji rešuje problem mešanega celoštevilskega linearnega programiranja (angl. Mixed-Integer Linear Programming, MILP), kjer nalogam za vsakega delavca in vsak dan posebej dodeli čas njihovega začetka. V tem prispevku se posvečamo študiji učinkovitosti algoritma na tretjem nivoju. Izkaže se, da ta nalog ne more razporediti dovolj hitro za praktično uporabo, zato za povečanje njegove učinkovitosti MILP poenostavimo. Rezultati poskusov kažejo, da poenostavitev izboljša učinkovitost algoritma na tretjem nivoju, medtem ko je učinek na celoten algoritem načeloma ugoden, a odvisen od problema.

KLJUČNE BESEDE

problem razporejanja, evolucijski algoritem, hevristika, algoritem razveji in omeji, mešano celoštevilsko linearno programiranje, učinkovitost

ABSTRACT

Fieldwork scheduling is a demanding optimization problem. To solve it, we developed a three-level algorithm. At the first level, an evolutionary algorithm distributes tasks to workers, at the second level, a heuristic algorithm distributes tasks of each worker over days, and at the third level, a branch-and-bound algorithm solves the problem in the form of mixed-integer linear programming (MILP), where the starting times of tasks need to be scheduled for each worker and each day separately. In this paper, we study the efficiency of the algorithm at the third level. Because it cannot schedule the tasks fast enough for practical use, we try to increase its efficiency by simplifying the MILP. Experimental results show that the simplification improves the performance of the algorithm at the third level, while the effect on the overall algorithm is in principle favorable, but depends on the problem.

KEYWORDS

scheduling problem, evolutionary algoirthm, heuristic algorithm, branch-and-bound algorithm, mixed-integer linear programming, efficiency

1 UVOD

Razporejanje terenskega dela je optimizacijski problem, ki zahteva dodelitev delavca in časa začetka opravljanja vsaki terenski nalogi tako, da je zadoščeno vsem omejitvam razporejanja in je cena celotnega urnika čim nižja. Obstajajo številne različice tega problema, ki se razlikujejo tako po omejitvah kot po načinu izračuna cene razporeda. Posledično obstajajo tudi različni pristopi za njegovo reševanje [6]. Študija [3] primerja dve formulaciji problema, in sicer v obliki problema usmerjanja vozil (angl. Vehicle Routing Problem, VRP) in v obliki problema mešanega celoštevilskega linearnega programiranja (angl. Mixed-Integer Linear Programming, MILP). Rezultati poskusov študije nakazujejo, da je oblika MILP za zapis razporejanja nalog terenskega dela ustreznejša od oblike VRP, zato tudi naš pristop uporablja obliko MILP.

Vendar pa je učinkovitost reševanja takšnih kombinatoričnih problemov zelo odvisna od njihove velikosti. Že pri relativno majhnih problemih se namreč pogosto zgodi, da jih ni moč rešiti v doglednem času. Zato se v našem pristopu zgledujemo po podobnih prijemih iz sorodnega dela (glej npr. [1]) in problem razdelimo na manjše, lažje obvladljive podprobleme. Problem razporejanja terenskega dela tako rešujemo s trinivojskim optimizacijskim algoritmom, pri katerem na prvem nivoju evolucijski algoritem razporedi naloge po delavcih, na drugem nivoju hevristika za vsakega delavca razporedi naloge po dnevih, na tretjem nivoju pa algoritem razveji in omeji rešuje problem v obliki MILP, tj. nalogam za vsakega delavca in vsak dan posebej dodeli čas njihovega začetka.

Tak trinivojski algoritem je sposoben v uri zadovoljivo rešiti tudi nekoliko večje probleme (npr. z 20 delavci, 20 dnevi in več sto nalogami), a je ta čas za praktično uporabo predolg. Zato želimo algoritem pohitriti. Ozko grlo predstavlja reševanje problema MILP, saj sta evolucijski algoritem in hevristika zelo hitra, tako da lahko največjo pohitritev celotnega algoritma dosežemo s pohitritvijo na tretjem nivoju.

V nadaljevanju prispevka v 2. razdelku najprej predstavimo našo različico problema razporejanja terenskega dela, nato pa v 3. razdelku na kratko opišemo trinivojski algoritem za njeno reševanje. V 4. razdelku analiziramo učinkovitost algoritma na tretjem nivoju, v 5. razdelku pa predlagamo poenostavitev problema MILP in preverimo njen učinek najprej na algoritem na

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia

tretjemu nivoju in končno na celoten trinivojski algoritem. Prispevek sklenemo z zaključki v 6. razdelku.

2 PROBLEM RAZPOREJANJA TERENSKEGA DELA

Problem razporejanja terenskega dela opišemo s scenarijem razporejanja, spremenljivkami problema, omejitvami in optimizacijskim kriterijem (podrobne formalne definicije tu ne moremo zapisati zaradi pomanjkanja prostora). Obravnavamo najbolj splošno različico problema, v kateri želimo razporediti večino nalog, saj ta pokriva tudi posebni primer, ko je zaradi spremembe v zadnjem trenutku treba prerazporediti samo nekaj nalog.

2.1 Scenarij razporejanja

Časovno obdobje razporejanja je razdeljeno na dneve, znotraj njih je čas obravnavan zvezno. Za vsak dan poznamo začetek in konec rednega delovnika ter trajanje morebitnih nadur (bodisi na začetku bodisi na koncu dneva). Dane imamo tudi množico *lokacij*, časovne oddaljenosti za vsak par lokacij ter množico *kompetenc*, ki so skupne nalogam in delavcem. Scenarij razporejanja vsebuje tudi podatke o *delavcih*, in sicer za vsakega kompetence, dovoljeno število nadur ter začetno in končno lokacijo. Podatki o *nalogah* pa za vsako obsegajo njeno trajanje, želeno in obvezno časovno okno, prioriteto, zahtevane kompetence in morebitne želene delavce. *Malice* so posebne naloge, za katere lokacija ni definirana (malica se vedno izvaja na isti lokaciji kot predhodna naloga in se ne more prekrivati z drugimi nalogami).

Dodatno lahko scenarij razporejanja vsebuje že vnaprej pripravljene razporede posameznih nalog, ki so dveh tipov. *Obveznih razporedov* se ne sme spreminjati, a jih je treba vseeno upoštevati, saj postavljajo omejitve k razporejanju ostalih nalog. Po drugi strani pa se *želene razporede* lahko spreminja, a to vpliva na ceno končnega urnika.

2.2 Spremenljivke

Spremenljivke optimizacijskega problema v celoti določijo urnik, saj za vsako nalogo povedo ali je razporejena ali ni (ni namreč treba razporediti vseh nalog) in če je, kateri delavec jo bo opravil ter kdaj se bo začela izvajati.

2.3 Omejitve

Urnik, ki predstavlja rešitev problema, je dopusten samo, če izpolnjuje vse naslednje omejitve:

- Delavec lahko izvaja samo eno nalogo hkrati (v časovnem razporejanju nalog je treba poskrbeti tudi za upoštevanje trajanja potovanja med lokacijami).
- Delavec lahko izvaja naloge le znotraj delovnega časa in ima omejeno število nadur.
- Delavec mora imeti zahtevane kompetence za opravljanje naloge.
- Naloge morajo biti razporejene znotraj svojih obveznih časovnih oken.
- Nalog z obveznim razporedom se ne sme prerazporejati.

2.4 Optimizacijski kriterij

Optimizacijski kriterij oz. cena urnika, ki jo želimo minimizirati, je definirana kot utežena vsota naslednjih delnih kriterijev (prve tri postavke so si v nasprotju, zato je smiselno upoštevati samo eno od njih naenkrat):

- Vsi delavci naj bodo čim bolj enakomerno obremenjeni.
- Dnevno aktivnih delavcev naj bo čim manj.
- Aktivni delavci naj bodo čim bolj enakomerno obremenjeni.
- Skupno trajanje potovanj med lokacijami naj bo čim krajše.
- Izvede naj se čim več nalog.
- Naloge naj se izvedejo čim prej.
- Delavci naj imajo čim manj neaktivnega časa.
- Nadur naj bo čim manj.
- Naloge z višjo prioriteto naj se začnejo izvajati pred nalogami z nižjo prioriteto.
- Naloge, ki zapadejo prej, naj se začnejo izvajati pred nalogami, ki zapadejo kasneje.
- Naloge naj se izvedejo čim bliže želenemu časovnemu oknu.
- Nalogo, ki ima želene delavce, naj opravi eden izmed želenih delavcev.
- Nalogo z želenim razporedom naj se izvede čim bliže temu razporedu.

Uteži posameznih delnih kriterijev so zelo pomembne, saj določajo njihova medsebojna razmerja in drastično vplivajo na dobljene rešitve. Nastavili smo jih s pomočjo ekspertnega znanja in poskusov na številnih različnih scenarijih.

3 TRINIVOJSKI OPTIMIZACIJSKI ALGORITEM

V nadaljevanju na kratko predstavimo vse tri nivoje optimizacijskega algoritma.

3.1 Prvi nivo: razporejanje nalog po delavcih

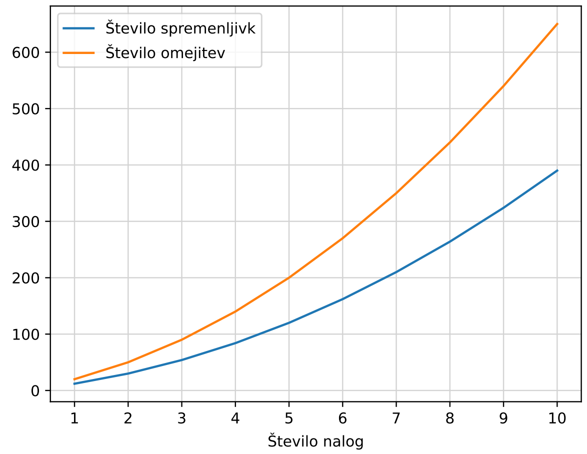
Na tem nivoju z evolucijskim algoritmom [5] vsaki nalogi dodelimo delavca, ki jo bo izvedel. Evolucijski algoritem začetno populacijo N_p rešitev ustvari naključno, vendar tako, da vse rešitve ustrezajo omejitvam za delavce (prve tri omejitve v razdelku 2.3). Potem algoritem izvaja naslednje korake največ N_g generacij. V vsaki generaciji algoritem najprej izbere N_p staršev s turnirsko selekcijo. Nato pare staršev križa in mutira (pri mutaciji uporabimo različne strategije zasnovane po meri delnih kriterijev (glej razdelek 2.4), ki jih izbiramo tako, da se pogostost uporabe sklada z njihovimi utežmi). Tako dobljeno populacijo evolucijski algoritem ovrednoti tako, da za vsako rešitev izvede drugi in tretji nivo algoritma. Nato staro populacijo prepiše z novo (najboljšo staro rešitev ohrani) in nadaljuje z enakimi koraki.

3.2 Drugi nivo: razporejanje nalog delavca po dnevih

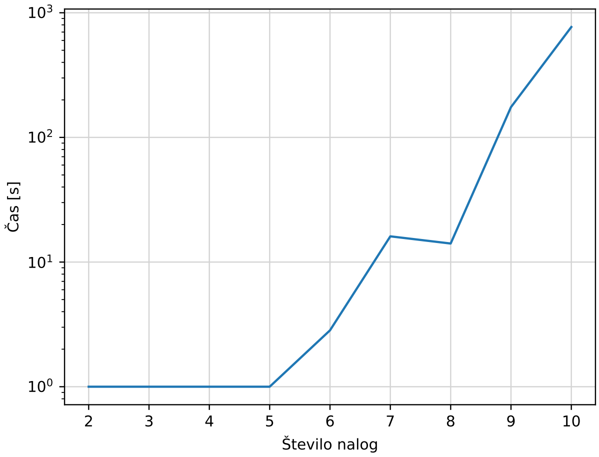
Hevristika na drugem nivoju naloge vsakega delavca razporedi po dnevih. Po vrsti vsem nalogam, urejenim naraščajoče po številu dni, v katerih se lahko izvedejo, dodelimo zanje najugodnejši dan. Ugodnost dne določimo z glasovanjem, ki poteka tako, da različni delni kriteriji (glej razdelek 2.4) glasujejo za dneve, ki so zanje najugodnejši. Glasovi so uteženi z utežmi delnih kriterijev, nalogi pa dodelimo dan z največ glasovi.

3.3 Tretji nivo: določitev časa začetka nalog za en dan enega delavca

Na tretjem nivoju z algoritmom razveji in omeji nalogam za en dan enega delavca dodelimo začetni interval. Problem torej zapišemo v obliki MILP tako, da upoštevamo samo tiste omejitve in delne kriterije, ki so na tem nivoju še smiselni (npr. na tem



Slika 1: Odvisnost števila spremenljivk in omejitev v formulaciji problema MILP na tretjem nivoju od števila nalog.



Slika 2: Povprečen čas, potreben za optimalno rešitev problema glede na njegovo velikost.

nivoju se ne ukvarjamo več s kompetencami, enakomerno obremenjenostjo delavcev in podobnimi delnimi kriteriji, saj je zanje poskrbljeno na prvih dveh nivojih).

Podobno kot pri predstavitvi problema tudi tu zaradi omejenega prostora ne moremo navesti celotne formulacije problema MILP. Za razumevanje nadaljevanja je najpomembneje vedeti, da imamo pri takšni formulaciji za problem z n nalogami $3n^2 + O(n)$ spremenljivk in $5n^2 + O(n)$ omejitev, kot prikazuje slika 1.

4 PREIZKUS UČINKOVITOSTI

Kot je razvidno iz slike 1, je število spremenljivk problema MILP zelo veliko že za probleme z majhnim številom nalog, kar otežuje nalogo optimizacijskemu algoritmu. Čas, ki ga potrebuje za najdbo optimalne rešitve, preverimo s poskusom na množici testnih problemov, ki imajo lastnosti podobne problemom iz prakse.

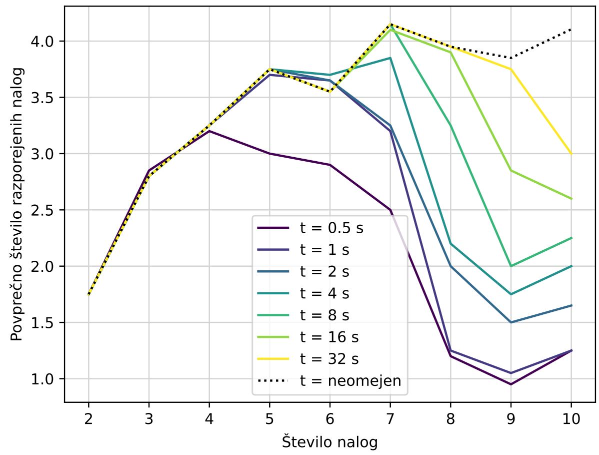
Ta množica vsebuje 180 testnih problemov (20 za vsako velikost problema od dveh do desetih nalog), pri katerih je treba določiti čas izvajanja nalog za enega delavca v enem dnevu. Nekateri problemi imajo samo navadne naloge, lahko pa imajo tudi malico, eno nalogo z obveznim razporedom ali pa oboje. Trajanje malice je vedno pol ure, trajanje ostalih nalog pa je izbrano naključno iz porazdelitve, ki skuša posnemati probleme iz prakse. Tako je večina nalog krajših od 90 minut, nekaj pa jih ima dolžino do štirih ur. Prioriteta vsake naloge je izbrana naključno med 1 in 9. Prav tako so lokacije izvajanja nalog in začetna lokacija delavca izbrane naključno izmed lokacij nekaterih večjih slovenskih mest. Večina nalog ima neomejeno časovno okno, pri nekaterih pa je okno skrajšano na začetku ali koncu dneva. Trajanje delovnega časa je izbrano naključno med 6 in 10 ur, lahko pa delavec vedno opravlja do dve naduri.

Vse testne probleme rešujemo z algoritmom razveji in omeji iz reševalnika SCIP [2] preko knjižnice OR-Tools [4]. Pri tem beležimo čas, ki ga algoritem potrebuje, da najde optimalno rešitev. Reševanje poteka na osebnem računalniku s 16 GB pomnilnika in frekvenco procesorja 3,60 GHz.

Rezultati poskusa so prikazani na sliki 2. Vidimo, da algoritem praviloma potrebuje eksponentno več časa z dodajanjem vsake naloge. Če želimo, da je celoten trinivojski algoritem koristen v praksi, si lahko za reševanje problema MILP privoščimo le nekaj sekund, kar pomeni, da je za naše potrebe algoritem neučinkovit že za probleme s sedmimi ali več nalogami.

Preverimo še, kako dobro deluje algoritem, če mu omejimo čas, ki ga ima na voljo za iskanje rešitev. Poskuse izvedemo z naslednjimi časovnimi omejitvami: 0.5 s, 1 s, 2 s, 4 s, 8 s, 16 s in 32 s. Pri tem opazujemo, koliko nalog je algoritem razporedil, in to število primerjamo z optimalnim številom razporejenih nalog (dobljenim v prejšnjem poskusu, ko algoritem ni bil časovno omejen). Čeprav cilj algoritma ni samo razporediti čim več nalog, je število razporejenih nalog dober pokazatelj kakovosti delovanja algoritma.

Na sliki 3 vidimo, da ob prekratkem času na problemih z veliko nalogami algoritem odpove (večino nalog zavrne, čeprav bi jih lahko razporedil). Na primer, ko ima algoritem na voljo le 0,5 s, primerno deluje le za probleme z do štirimi nalogami, za večje problema pa njegova uspešnost pade in ko je nalog osem ali več, v povprečju razporedi le eno nalogo. Delovanje algoritma je nekoliko boljše, če ima na voljo daljši čas, a šele pri 32 s se



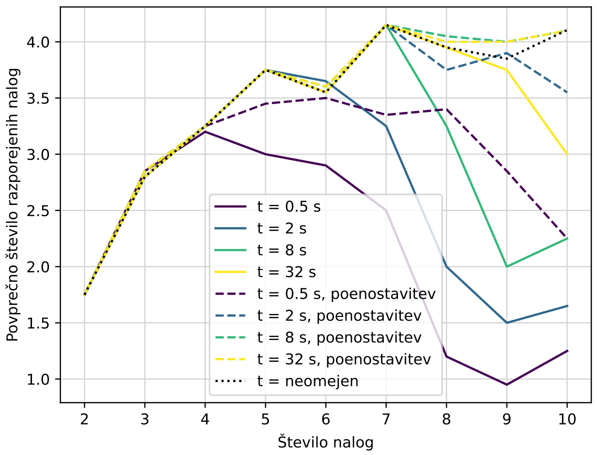
Slika 3: Povprečno število razporejenih nalog pri različnih časovnih omejitvah in velikostih problemov (s črtkano črno črto je prikazano optimalno število razporejenih nalog).

število razporejenih nalog na problemih z devetimi in desetimi nalogami približa optimalnemu številu razporejenih nalog.

5 POENOSTAVITEV PROBLEMA

Ker z delovanjem algoritma nismo zadovoljni, poskusimo problem poenostaviti. Za zapis delnih kriterijev za prioriteto in čas zapadlosti potrebujemo $2n^2 + O(n)$ spremenljivk ter $4n^2 + O(n)$ omejitev, kar je zelo veliko, sploh ker ta dva delna kriterija nista zelo pomembna. Zato preizkusimo, kako algoritem deluje, če ju izpustimo (zanju lahko do neke mere poskrbimo na zgornjih dveh nivojih optimizacijskega algoritma). Število spremenljivk in omejitev se še vedno povečuje kvadratično s številom nalog, vendar pa smo koeficienta pred kvadratnim členom s 3 oziroma 5 zmanjšali na 1.

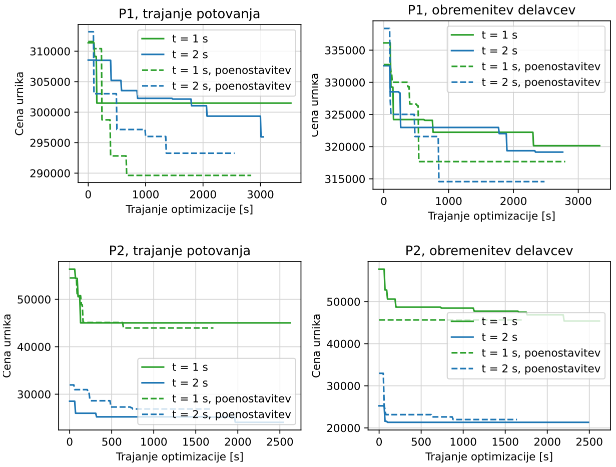
Za poenostavljeni problem izvedemo podoben test kot v razdelku 4, le da testiramo samo pri časovnih omejitvah 0.5 s, 2 s, 8 s in 32 s. Na sliki 4 primerjamo število razporejenih nalog pri osnovnemu ter poenostavljenemu problemu. Vidimo, da na večjih poenostavljenih problemih algoritem deluje mnogo bolje.



Slika 4: Primerjava delovanja algoritma na izvirni (polne črte) in poenostavljeni formulaciji problema (črtkane črte) pri različnih časovnih omejitvah in velikostih problemov (s črno črtkano črto je prikazano optimalno število razporejenih nalog izvirne formulacije problema).

S poenostavitvijo torej dosežemo, da algoritem na tretjem nivoju deluje zadovoljivo tudi za praktične potrebe. Vendar pa to še ne pomeni nujno, da poenostavitev izboljša delovanje celotnega trinivojskega algoritma. To preverimo s poskusom na dveh testnih problemih, P1 in P2, pri katerih damo enkrat večjo utež delnemu kriteriju trajanja potovanja, drugič pa enakomerni obremenitvi delavcev. Na ta način dobimo štiri različne testne probleme. P1 obsega 220 nalog, deset delavcev in sedem dni, P2 pa 114 nalog (vsebuje tudi malice), pet delavcev in tri dni. Za vsak testni problem poženemo štiri različice trinivojskega algoritma, ki se razlikujejo samo na tretjem nivoju – ta uporablja bodisi izvirni bodisi poenostavljeni problem, izvajanje algoritma pa je omejeno bodisi na 1 s bodisi na 2 s.

Slika 5 kaže rezultate teh poskusov. Na problemu P1 (zgornja dva grafa na sliki) lahko jasno vidimo, da poenostavitev problema na tretjem nivoju koristi učinkovitosti celotnega algoritma. Tega ne moremo trditi za problem P2, na katerem je delovanje izvirne in ponostavljene različice zelo podobno, vidimo pa veliko boljše delovanje v primeru omejitve izvajanja na 2 s.



Slika 5: Rezultati optimizacije za štiri različice algoritma na dveh problemih (P1 zgoraj in P2 spodaj) z dvema različnima delnima kriterijema (trajanje potovanja levo in enakomerna obremenitev delavcev desno). Manjše vrednosti so boljše.

6 ZAKLJUČKI

V prispevku smo analizirali učinkovitost algoritma za razporejanje terenskega dela. Posvetili smo se le časovno najzahtevnejšemu delu trinivojskega algoritma – reševanju problema MILP na tretjem nivoju. Z dvema poskusoma smo pokazali, da algoritem razveji in omeji ni dovolj učinkovit za reševanje praktičnih problemov, zato smo problem MILP poenostavili. To ne spremeni kriterijev celotnega problema, algoritmu na tretjem nivoju pa omogoči, da učinkovito reši tudi probleme z desetimi nalogami (več jih v praksi ne pričakujemo). Primerjali smo tudi, kako poenostavitev vpliva na delovanje celotnega algoritma, in ugotovili, da čeprav obstajajo problemi, za katere poenostavitev ni koristna, v splošnem daje dobre rezultate in se je bomo posluževali tudi v praksi.

ZAHVALA

To delo je nastalo v okviru projekta Inteligentno in okolju prijazno razporejanje terenskega dela – MF-Scheduler, katerega naročnik je Comland d.o.o., sofinancerja pa Ministrstvo za gospodarski razvoj in tehnologijo Republike Slovenije in Evropski sklad za regionalni razvoj Evropske unije, ter raziskovalnega programa P2-0209, ki ga financira Javna agencija za raziskovalno dejavnost Republike Slovenije iz državnega proračuna.

LITERATURA

[1] S. Bertels in T. Fahle. 2006. A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem. *Computers & Operations Research*, 33, 10, 2866–2890. DOI: 10.1016/j.cor.2005.01.015.

[2] K. Bestuzheva in sod. 2021. The SCIP Optimization Suite 8.0. Technical Report. Optimization Online, (dec. 2021). http://www.optimization-online.org/DB_HTML/2021/12/8728.html.

[3] J. A. Castillo-Salazar, D. Landa-Silva in R. Qu. 2016. Workforce scheduling and routing problems: Literature survey and computational study. *Annals of Operations Research*, 239, 1, 39–67. DOI: 10.1007/s10479-014-1687-2.

[4] Google Developers. 2021. About OR-Tools. Retrieved 19. avg. 2022 from https://developers.google.com/optimization/introduction/overview.

[5] A. E. Eiben in J. E. Smith. 2015. *Introduction to Evolutionary Computing*. (2. izd.). Springer. DOI: 10.1007/978-3-662-44874-8.

[6] D. C. Paraskevopoulos, G. Laporte, P. P. Repoussis in C. D. Tarantilis. 2017. Resource constrained routing and scheduling: Review and research prospects. *European Journal of Operational Research*, 263, 3, 737–754. DOI: 10.1016/j.ejor.2017.05.035.

Interaktivno eksperimentiranje z besednimi vložitvami v platformi ClowdFlows

Interactive Experimentation with Word Embeddings in the ClowdFlows platform

Martin Žnidaršič
martin.znidarsic@ijs.si

Senja Pollak
senja.pollak@ijs.si

Vid Podpečan
vid.podpecan@ijs.si

Institut "Jožef Stefan"
Jamova cesta 39
1000 Ljubljana, Slovenija

POVZETEK

V članku predstavimo spletno platformo ClowdFlows, ki je namenjena analiziranju podatkov in strojnemu učenju in omogoča uporabo interaktivnih delotokov. Posebej predstavimo značilnosti platforme, ki lajšajo njeno uporabo programiranja neveščim uporabnikom in elemente platforme, ki omogočajo analizo teksta z najsodobnejšimi pristopi vektorskih vložitev. Poročamo tudi o praktičnem preizkusu uporabnosti platforme in njenih orodij z vektorskimi vložitvami za izbrane ciljne uporabnike s področij humanistike in družboslovja.

KLJUČNE BESEDE

procesiranje naravnega jezika, besedne vložitve, spletna aplikacija, delotoki

ABSTRACT

The paper presents the ClowdFlows web platform for machine learning and data analysis using interactive workflows. In particular, we highlight selected features that facilitate its use by non-programmers as well as selected elements of the platform that enable text analysis using state-of-the-art word embedding approaches. We also report on a hands-on evaluation of the usability of the platform and its word embedding components in a selected group of end users from the fields of humanities and social sciences.

KEYWORDS

natural language processing, word embeddings, web application, workflows

1 UVOD

Področja, povezana z metodami umetne inteligence, kot so rudarjenje podatkov, strojno učenje in avtomatska obdelava naravnega jezika, v zadnjih letih doživljajo razmah v praktični uporabi. Najnovejši metodološki dosežki so običajno najprej na voljo v obliki programskih knjižnic ali spletnih storitev (angl. *web services*), pozneje v platformah za razvijanje rešitev z udobnim uporabniškim vmesnikom in običajno še pozneje v namenskih orodjih, ki to metodologijo uporabljajo interno in omogočajo njeno uporabo brez ali z zelo omejenim vplivom na način delovanja tudi uporabnikom brez računalniškega predznanja. Slednjim samostojno rabo tovrstnih metod med drugim otežuje potrebno predznanje, ki je potrebno za njihovo smiselno uporabo, včasih pa tudi postopki namestitve in nastavitev programske opreme. Prototipno raziskovalno orodje ClowdFlows, ki ga razvijamo na Odseku za tehnologije znanja na Institutu "Jožef Stefan", naslavlja ti dve oviri in kaže potencial za praktično uporabo. V sklopu projekta EMBEDDIA [14, 13, 16] smo razširili nabor zmogljivosti tega orodja predvsem na področju analize naravnega jezika, zato se v tem prispevku osredotočamo na metode in končne uporabnike s tega področja. Natančneje, predstavimo primer učenja in uporabe modelov za besedne vektorske vložitve in izkušnje novih uporabnikov s področja humanistike in družboslovja.

V razdelku 2 predstavimo osnovno sorodno delo. Platforma ClowdFlows je opisana v razdelku 3. Razdelek 4 predstavi primer uporabe vektorskih vložitev in uporabniške izkušnje. Zaključki so podani v razdelku 5.

2 OZADJE IN SORODNO DELO

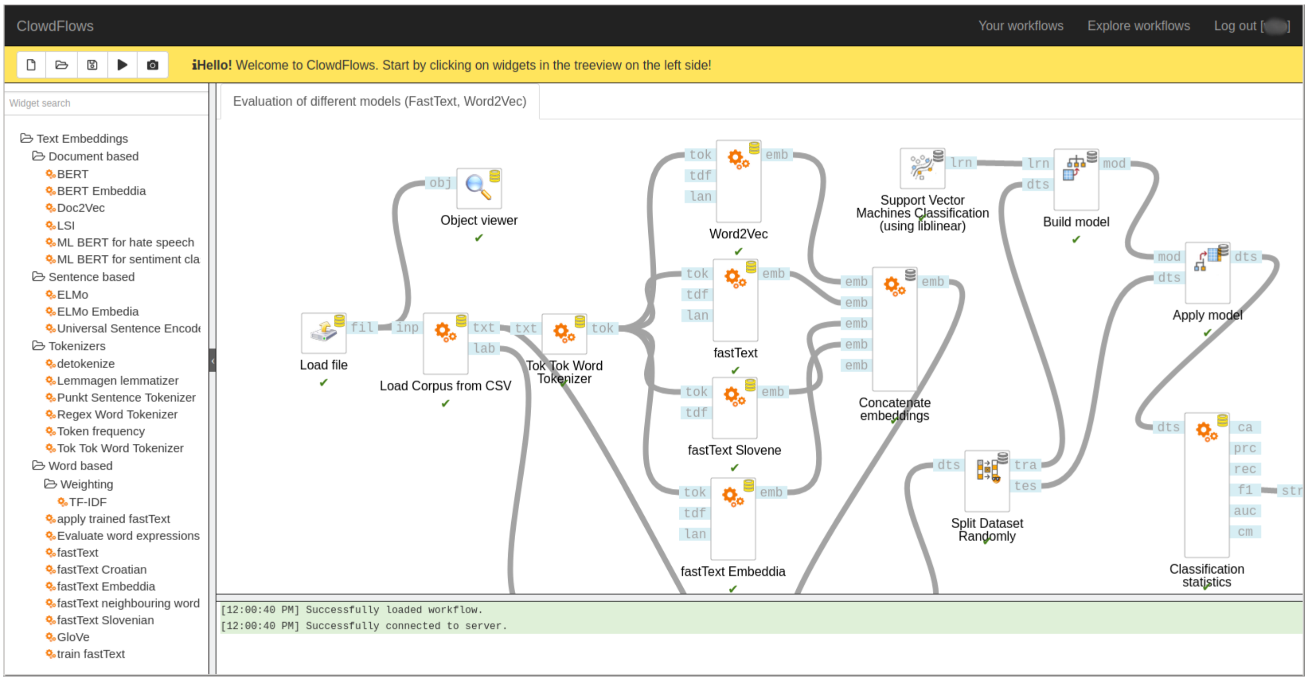
2.1 Platforme za vizualno programiranje in deljenje rešitev

Programsko orodje ClowdFlows, ki je predstavljeno in uporabljeno v tem prispevku, je podobno nekaterim drugim orodjem za upravljanje delotokov podatkovnega rudarjenja. Slovenskim uporabnikom je verjetno najbolj poznano orodje Orange [2], podobni pa sta orodji tudi Weka [18] in RapidMiner [8, 5]. Vsa ta orodja omogočajo vizualno programiranje s programskimi gradniki in upravljanje tako izdelanih programov. Manj razširjene so rešitve za skupno rabo delovnih tokov. To recimo ponuja portal myExperiment [15] ali spletna stran pobude OpenML [17]. Je pa uporabnost teh rešitev omejena predvsem na dobro podprto javno deljenje rešitev, za izvajanje ali urejanje delovnih tokov pa mora uporabnik še vedno namestiti posebno programsko opremo, v kateri so bili le-ti zasnovani. ClowdFlows, po drugi strani, omogoča tako izdelavo kot tudi deljenje in izvajanje delotokov.

2.2 Besedne vložitve

Besedne vektorske vložitve, ki so strojno naučene z uporabo nevronskih mrež, so predstavitve besed v prostoru, kjer vsako besedo opisuje vektor z veliko dimenzijami (tipično od nekaj deset do nekaj sto). Besede, ki so si blizu v vektorskem prostoru (kar lahko merimo s kosinusno razdaljo), so si tudi semantično podobne. Med vektorskimi vložitvami je mogoče računati tudi odnose, ki presegajo enostavno sorodnost besed, npr. preko analogij. Na primer, odnos *Madrid:Španija* je podoben odnosu *Pariz:Francija* [10]. Pri statičnih vložitvah, kot so modeli word2vec [9] in fastText [1], je posamezna beseda v korpusu predstavljena z enim vektorjem. Pri metodi fastText je vsaka

Information Society 2022, 10–14 October 2022, Ljubljana, Slovenia



¡Hello! Welcome to ClowdFlows. Start by clicking on widgets in the treeview on the left side!

```
[12:00:40 PM] Successfully loaded workflow.
[12:00:40 PM] Successfully connected to server.
```

Slika 1: Glavni pogled v ClowdFlows.

beseda predstavljena kot vsota vektorskih vložitev znakovnih n-gramov, ki jih beseda vsebuje. V praksi to pomeni, da metoda pri modeliranju semantične bližine upošteva tudi morfološko podobnost besed, zaradi česar je ta metoda še posebej uporabna za izračun besednih vložitev v morfološko bogatih jezikih, kot je slovenščina. Za razliko od statičnih vložitev pa pri kontekstualnih vložitvah, kot sta na primer modela ELMo [12] in BERT [3], vsako pojavitev besede opisuje svoj vektor. To je pomembno predvsem z vidika večpomenskih besed pa tudi v primerih, kjer analiziramo razlike med besedami v različnih kontekstih. Za veliko jezikov obstajajo prednaučeni modeli na velikih jezikovnih korpusih [4, 3], ki jih je mogoče priučiti za posamezne domene in naloge.

3 CLOWDFLOWS

ClowdFlows [6, 7] je spletna platforma za analiziranje podatkov in strojno učenje z grafičnim uporabniškim vmesnikom, ki omogoča izvajanje v brskalniku brez zahtev po lokalni namestitvi programske opreme, ponuja pa tudi preprosto javno deljenje izdelanih rešitev. Gre za odprtokodno raziskovalno orodje, katerega zadnja stabilna različica ClowdFlows 3 je na voljo na naslovu: `https://cf3.ijs.si/`.

Grafičen način sestave delovnih tokov in uporaba javno deljenih rešitev brez nameščanja dodatne programske opreme sta značilnosti, ki lajšata uporabo tudi uporabnikom, ki nimajo programerskega predznanja, imajo pa zanimive podatke in raziskovalne probleme, pri katerih bi jim prav prišle metode, ki so na voljo v ClowdFlows. Za raziskovalce je poleg tega pomembno tudi preprosto deljenje in preprostost ponavljanja ali nadgrajevanja obstoječih eksperimentov.

Elementi v ClowdFlows 3 vsebujejo vrsto programskih gradnikov, ki ponujajo delo z vektorskimi vložitvami. Vsebujejo prednaučene statične in kontekstualne modele za več jezikov kakor tudi nekaj orodij, ki na njih temeljijo, kot so na primer klasifikatorji za analizo sentimenta novic [11] in prepoznavanje sovražnega govora [11].

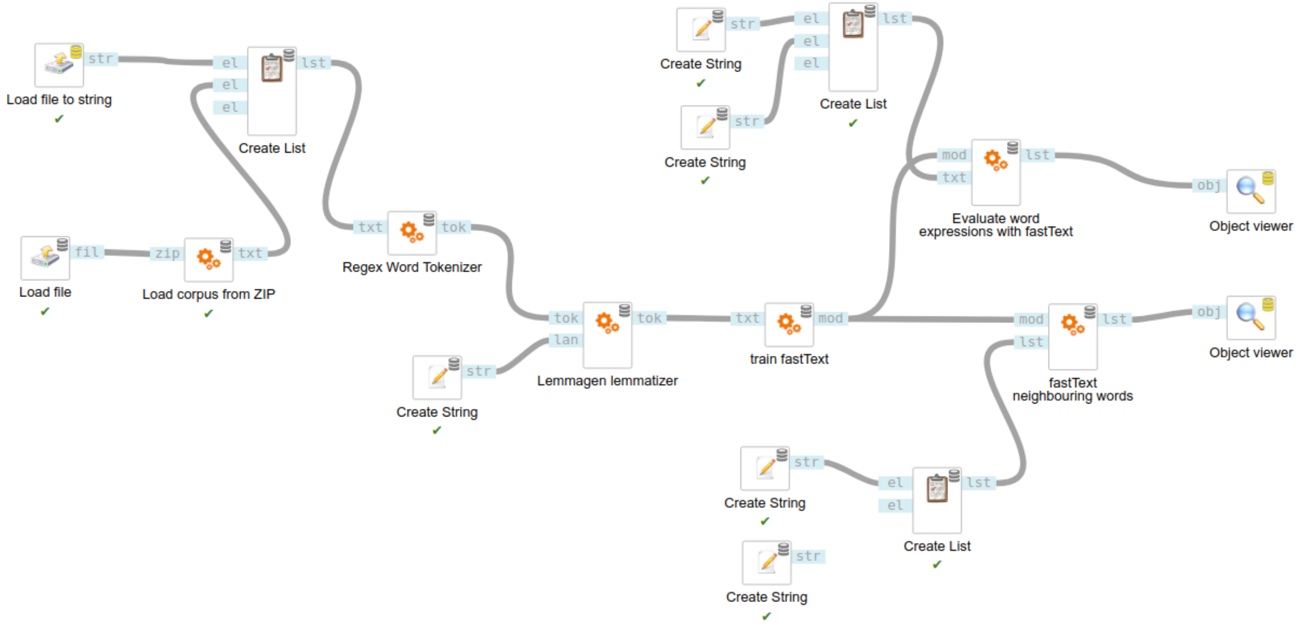
Po prijavi v ClowdFlows imamo na voljo kratek tečaj o osnovah, izdelavo novega delotoka ali pregled javno dostopnih rešitev. Glavni pogled je namenjen izdelavi, pregledu in poganjanju delotokov. Prikazan je na sliki 1. Večji del tega pogleda predstavlja delovna površina, na katero lahko potegnemo (ali uvrstimo z dvoklikom) željeni programski gradnik (angl. *widget*) iz seznama razpoložljivih gradnikov na levi strani pogleda. Smiselno povezani gradniki predstavljajo delotok, ki ga lahko poženemo z nadzornim gumbom *Play*.

Povezave med gradniki vzpostavimo s klikom na izhod enega gradnika in vhod drugega. Vhodi so predstavljeni kot svetlo modri pravokotniki na levi strani gradnika in izhodi kot tovrstni pravokotniki na desni. Povezave lahko odstranimo tako, da z desno tipko miške kliknemo povezavo in izberemo možnost *Remove*. Delotoki se shranjujejo samodejno, lahko pa jih tudi eksplicitno shranimo s pritiskom na kontrolnik za shranjevanje, kar nam omogoča tudi lastno poimenovanje shranjenega dela. Shranjene delotoke lahko pregledujemo, kopiramo, brišemo, izvažamo ali javno delimo na pogledu, ki se pokaže ob izbiri *Your workflows*. Javno objavljeni delotoki dobijo nespremenljiv URL naslov, ki ga lahko delimo in vsakemu uporabniku ClowdFlows omogoča, da ustvari svojo kopijo tako deljenega dela.

4 UPORABA VEKTORSKIH VLOŽITEV

4.1 Učenje modela vložitev v ClowdFlows

Za pridobitev predstavitve teksta v obliki vektorskih vložitev lahko uporabimo predpripravljene modele ali pa take modele sami strojno naučimo. Tovrstno strojno učenje je običajno računsko zelo zahtevno in za smiselne rezultate potrebuje velike količine podatkov. V praksi se zato pogosto uporablja predhodno naučene modele. ClowdFlows ponuja več prednaučenih modelov, naprimer ELMo, Word2Vec in razne modele pristopa BERT in fastText, tudi za slovenščino.



Slika 2: Delotok, ki je bil uporabljen na delavnici s ciljnimi uporabniki. Dostopen je na: https://cf3.ijs.si/workflow/283

.

Učenje lastnih modelov je smiselno, ko gre za posebna besedila ali naloge, pri katerih jih želimo uporabljati. Velika računska zahtevnost, velike količine podatkov in z njima povezani daljši časi obdelave namreč niso združljivi z interaktivno uporabo, ki je značilna za ClowdFlows. Pri uporabnikih digitalne humanistike in družboslovja smo zaznali potrebo za učenje vložitev na majhnih, specifičnih korpusih, kot so pesniške zbirke, specializirani novičarski članki ipd. Takšni korpusi so pogosto bistveno manjši od tipičnih korpusov, ki se uporabljajo za učenje vektorskih vložitvev. Glede na potrebe uporabnikov in zmogljivosti platforme smo se odločili za implementacijo gradnika za učenje modelov *train fastText*, saj je algoritem *fastText* eden najučinkovitejših in najmanj računsko zahtevnih. Implementacija gradnika v ClowdFlows vsebuje tudi namige, kako prilagoditi privzete parametre za učenje na majhnih korpusih. Za sprejemljivo hitro interaktivno delo vseeno priporočamo, da vhodni korpus ne presega dveh milijonov besed ali približno 10 MB neobdelanega besedila.

Gradnik *train fastText* z uporabo algoritma fastText nauči nov vektorski model na vhodnem korpusu. Tak model lahko nato posredujemo drugim gradnikom. Vhod v *train fastText* je besedilni korpus, kot je na primer izhod gradnika *Load Corpus from CSV*. Korpus je mogoče tokenizirati, lematizirati ali pa uporabiti tudi brez tovrstne predobdelave.

train fastText uporabniku ponuja nastavljanje sledečih parametrov:

bucket - število skupin (značilke besednih in znakovnih *n*-gramov so zgoščene v fiksno število skupin);
epoch - število epoh učenja;
lr - hitrost učenja;
dimension - velikost besednih vektorjev;
window - velikost kontekstnega okna;
model - vrsta nenadzorovanega fastText modela (cbow ali skipgram) ter
min_count - najmanjše število pojavitev besede, pri katerem se beseda še upošteva.

Kjer je primerno, opis parametra vključuje namig, ali je v primeru majhnih učnih podatkov priporočljivo povečati oz. zmanjšati vrednost parametra.

4.2 Izkušnje uporabnikov

Uporabnost platforme ClowdFlows in najpomembnejših komponent za analizo naravnega jezika z vidika ciljnih končnih uporabnikov smo preverjali v okviru enodnevne delavnice, ki je potekala (na daljavo) 27. januarja 2022. Delavnica je bila namenjena eni od naših primarnih ciljnih skupin: raziskovalcem z različnih področij humanistike in družboslovja, ki (predvidoma) niso vešči programiranja.

Za potrebe delavnice smo pripravili primer delotoka za analiziranje besedil z vektorskimi vložitvami. Prikazan je na sliki 2. Delotok se začne z dvema možnima načinoma vnosa vhodnih podatkov, nadaljuje z opcijsko uporabo tokenizatorja in lematizatorja (ta kot vhodni podatek sprejema tudi oznako jezika), čemur sledi učenje modela fastText. Naučeni model nato v delotoku uporabimo na dva načina: v gradniku *fastText neighboring words* pregledujemo okolico (sosednje besede) izbranih besed, v gradniku *Evaluate word expressions with fastText* pa na modelu preizkušamo uporabo izrazov (seštevanje, odštevanje) na vektorskih predstavitvah besed. Ogled rezultatov v obeh primerih omogočimo z gradnikom *Object viewer*.

Delavnica se je začela s skupno uvodno predstavitvijo platforme ClowdFlows in primera delotoka s slike 2, ki je trajala 20 minut in v kateri smo izbrane primere prikazali z uporabo besedila novele *Deseti brat* Josipa Jurčiča.

Temu je sledilo osem 20-minutnih sej, v katerih je vsak uporabnik ustvaril svoj primerek delotoka, naložil svoj korpus in preizkusil izbrane komponente ClowdFlows. Ena seja je bila namenjena enemu uporabniku in njegovim podatkom, drugi uporabniki pa so lahko prisostvovali kot opazovalci. Uporabnikom smo pri njihovem delu pomagali, če so imeli težave pri uporabi platforme ali pri pripravi svojih vhodnih podatkov. Udeležba na delavnici je bila na povabilo. Udeleženci, ki so bili povabljeni na delavnico, so raziskovalci s področij literarnih ved, sociologije, socialnega dela in sorodnih področij. Pripravili so lastne korpuse s svojih področij, kot so na primer tematski korpusi migracij, korpus del slovenskih literatov, korpus francoske poezije, LGBT, novice, ki govorijo o socialnem delu in podobno. Nekateri udeleženci so bili vabljeni v okviru interdisciplinarnih projektov SOVRAG in CANDAS. Nihče od udeležencev pa ni imel predhodnih izkušenj s ClowdFlows. Zaradi velikega zanimanja smo število sej povečali s predvidenih 8 na 10.

Uporabljeni korpusi so bili zelo raznoliki, udeležence pa so zanimali različni vidiki obdelave besedil. V večini primerov so bili začetnemu delotoku dodani dodatni gradniki, da bi rešili določeno težavo ali zadovoljili posebne interese. Udeleženci so na primer iskali podobnosti in razlike v sosedstvu besed na podlagi korpusov iz različnih obdobij ali od različnih avtorjev. Zanimale so jih tudi osnovne značilnosti takih korpusov, kot so recimo najpogosteje uporabljene besede, s čimer so bile povezane tudi druge osnovne operacije, kot je na primer filtriranje besed. Med delavnico sta bili odkriti dve specifični tehnični težavi: (I) napake so se pojavile v primeru vnosa besedila s posebnimi znaki, ki ni bilo kodirano v kodni tabeli UTF, in (II) nekateri gradniki, ki vsebujejo klice na spletne storitve, so poročali o preseženi časovni omejitvi.

Za udeležence je bil pripravljen anonimen vprašalnik, povezava do vprašalnika pa je bila posredovana po delavnici. Večini udeležencev se je prikazani potek dela zdel zelo uporaben. Velika večina (80%) še nikoli ni poskusila uporabljati vektorskih vložitev. O uporabniškem vmesniku ClowdFlows so večinoma poročali kot o preprostem za uporabo (preprost: 60%, zelo preprost: 30%), le enemu udeležencu pa se je zdel zapleten. Te rezultate je sicer treba upoštevati v kontekstu dejstva, da so odzivi zbrani kmalu po uporabi ClowdFlows, pri čemer je bila na voljo pomoč. Brez uvoda in pomoči odgovori morda ne bi bili tako pozitivni, vendar tega še nismo preizkusili. Večina udeležencev je menila, da bi ponovno uporabili ClowdFlows, če bi jim zagotovili vnaprej pripravljen delotok za njihov problem.

5 ZAKLJUČEK

Predstavili smo spletno platformo ClowdFlows, izbrane elemente, ki omogočajo napredno uporabo pristopov za učenje besednih vektorskih vložitev, in izkušnje nekaterih od naših ciljnih uporabnikov teh orodij.

Eden od ciljev platforme ClowdFlows je približanje uporabe najnovejših metod analize podatkov in strojnega učenja tudi uporabnikom, ki niso vešči programiranja. Izkušnje naše delavnice z nekaterimi od potencialnih uporabnikov so pokazale, da je to vsekakor smiselno, saj so na podlagi predpripravljenih delotokov uporabniki (sicer strokovnjaki na drugih področjih) lahko opravili analize na lastnih podatkih in tudi že iskali in predlagali nadaljnje postopke analiz, ki so smiselni in uporabni pri njihovem delu. Poleg metodoloških razširitev in tehničnih izboljšav platforme bomo zato v bodoče več pozornosti namenjali tudi razvoju primerov rešitev za ciljne uporabnike, v prvi vrsti za raziskovalce s področij, ki niso povezana z računalništvom.

ZAHVALA

Prispevek je rezultat raziskovalnih projektov *Računalniško podprta večjezična analiza novičarskega diskurza s kontekstualnimi besednimi vložitvami* (št. J6-2581), *Sovražni govor v sodobnih konceptualizacijah nacionalizma, rasizma, spola in migracij* (št. J5-3102) in programa *Tehnologije znanja* (št. P2-0103), ki jih je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije iz državnega proračuna, ter evropskega projekta EMBEDDIA (št. 825153), ki ga v okviru okvirnega programa za raziskave in inovacije Obzorje 2020 financira EU.

LITERATURA

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin in Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.

[2] Janez Demšar in sod. 2013. Orange: data mining toolbox in python. *Journal of Machine Learning Research*, 14, 1, 2349–2353.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee in Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. V *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

[4] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin in Tomas Mikolov. 2018. Learning word vectors for 157 languages. V *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[5] Markus Hofmann in Ralf Klinkenberg. 2016. *RapidMiner: Data mining use cases and business analytics applications*. CRC Press.

[6] Janez Kranjc. 2017. *Web Workflows for Data Mining in the Cloud*. Doktorska disertacija. Jožef Stefan International Postgraduate School.

[7] Janez Kranjc, Vid Podpečan in Nada Lavrač. 2012. ClowdFlows: a cloud based scientific workflow platform. V *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science. Zv. 7524. Peter A. Flach, Tijl Bie in Nello Cristianini, uredniki. Springer Berlin Heidelberg, 816–819.

[8] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz in Timm Euler. 2006. YALE: rapid prototyping for complex data mining tasks. V *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 935–940.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado in Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. V *Advances in Neural Information Processing Systems*, 3111–3119.

[10] Tomas Mikolov, Wen-tau Yih in Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. V *Proceedings of the 2013 Conference of the North American Chapter of the ACL: Human Language Technologies*. ACL, 746–751.

[11] Andraž Pelicon, Ravi Shekhar, Matej Martinc, Blaž Škrlj, Matthew Purver in Senja Pollak. 2021. Zero-shot cross-lingual content filtering: offensive language and hate speech detection. V *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Association for Computational Linguistics, Online, (apr. 2021), 30–34. https://aclanthology.org/2021.hackashop-1.5.

[12] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee in Luke Zettlemoyer. 2018. Deep contextualized word representations. V *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237.

[13] Senja Pollak in Andraž Pelicon. 2022. EMBEDDIA project: cross-lingual embeddings for less- represented languages in European news media. V *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation, Ghent, Belgium, (jun. 2022), 293–294.

[14] Senja Pollak in sod. 2021. EMBEDDIA tools, datasets and challenges: resources and hackathon contributions. V *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*. Association for Computational Linguistics, Online, (apr. 2021), 99–109. https://aclanthology.org/2021.hackashop-1.14.

[15] David De Roure, Carole Goble in Robert Stevens. 2009. The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows. *Future Generation Computer Systems*, 25, (feb. 2009), 561–567.

[16] Matej Ulčar, Aleš Žagar, Carlos S. Armendariz, Andraž Repar, Senja Pollak, Matthew Purver in Marko Robnik-Šikonja. 2021. Evaluation of contextual embeddings on less-resourced languages. (2021). https://arxiv.org/abs/2107.10614.

[17] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl in Luis Torgo. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15, 2, 49–60.

[18] Ian H Witten in Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Indeks avtorjev / Author index

Slovenska konferenca o umetni inteligenci

Slovenian Conference on Artificial Intelligence

Uredniki • Editors:
Mitja Luštrek, Matjaž Gams, Rok Piltaver