INFORMACIJSKA DRUŽBA

Zbornik 26. mednarodne multikonference

Zvezek C

INFORMATION SOCIETY

Proceedings of the 26th International Multiconference

Volume C

Odkrivanje znanja in podatkovna skladišča • SiKDD

Data Mining and Data Warehouses • SiKDD

Urednika • Editors:
Dunja Mladenić, Marko Grobelnik

9. oktober 2023 | Ljubljana, Slovenija • 9 October 2023 | Ljubljana, Slovenia

IS2023

Zbornik 26. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2023

Zvezek C

Proceedings of the 26th International Multiconference

INFORMATION SOCIETY – IS 2023

Volume C

Odkrivanje znanja in podatkovna skladišča – SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

9. oktober 2023 / 9 October 2023
Ljubljana, Slovenia

Urednika:

Dunja Mladenić
Department for Artificial Intelligence
Jožef Stefan Institute, Ljubljana

Marko Grobelnik
Department for Artificial Intelligence
Jožef Stefan Institute, Ljubljana

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

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

Ljubljana, oktober 2023

Informacijska družba
ISSN 2630-371X

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID **170733315**
ISBN 978-961-264-276-1 (PDF)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2023

Šestindvajseta multikonferenca Informacijska družba se odvija v obdobju izjemnega razvoja za umetno inteligenco, računalništvo in informatiko, za celotno informacijsko družbo. Generativna umetna inteligenca je s programi kot ChatGPT dosegla izjemen napredek na poti k superinteligenci, k singularnosti in razcvetu človeške civilizacije. Uresničujejo se napovedi strokovnjakov, da bodo omenjena področna ključna za obstoj in razvoj človeštva, zato moramo pozornost usmeriti na njih, jih hitro uvesti v osnovno in srednje šolstvo in vsakdan posameznika in skupnosti.

Po drugi strani se poleg lažnih novic pojavljajo tudi lažne enciklopedije, lažne znanosti ter »ploščate Zemlje«, nadaljuje se zapostavljanje znanstvenih spoznanj, metod, zmanjševanje človekovih pravic in družbenih vrednot. Na vseh nas je, da izzive današnjice primerno obravnavamo, predvsem pa pomagamo pri uvajanju znanstvenih spoznanj in razčiščevanju zmot. Ena pogosto omenjanih v zadnjem letu je eksistencialna nevarnost umetne inteligence, ki naj bi ogrožala človeštvo tako kot jedrske vojne. Hkrati pa nihče ne poda vsaj za silo smiselnega scenarija, kako naj bi se to zgodilo – recimo, kako naj bi 100x pametnejši GPT ogrozil ljudi.

Letošnja konferenca poleg čisto tehnoloških izpostavlja pomembne integralne teme, kot so okolje, zdravstvo, politika depopulacije, ter rešitve, ki jih za skoraj vse probleme prinaša umetna inteligenca. V takšnem okolju je ključnega pomena poglobljena analiza in diskurz, ki lahko oblikujeta najboljše pristope k upravljanju in izkoriščanju tehnologij. Imamo veliko srečo, da gostimo vrsto izjemnih mislecev, znanstvenikov in strokovnjakov, ki skupaj v delovnem in akademsko odprtem okolju prinašajo bogastvo znanja in dialoga. Verjamemo, da je njihova prisotnost in udeležba ključna za oblikovanje bolj inkluzivne, varne in trajnostne informacijske družbe. Za razcvet.

Letos smo v multikonferenco povezali deset odličnih neodvisnih konferenc, med njimi »Legende računalništva«, s katero postavljamo nov mehanizem promocije informacijske družbe. IS 2023 zajema okoli 160 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic, skupaj pa se je konference udeležilo okrog 500 udeležencev. 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 2023 sestavljajo naslednje samostojne konference:

- Odkrivanje znanja in podatkovna središča
- Demografske in družinske analize
- Legende računalništva in informatike
- Konferenca o zdravi dolgoživosti
- Miti in resnice o varovanju okolja
- Mednarodna konferenca o prenosu tehnologij
- Digitalna vključenost v informacijski družbi – DIGIN 2023
- Slovenska konferenca o umetni inteligenci + DATASCIENCE
- Kognitivna znanost
- Vzgoja in izobraževanje v informacijski družbi
- Zaključna svečana prireditev konference

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi ACM Slovenija, SLAIS za umetno inteligenco, DKZ za kognitivno znanost in 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. Andrej Brodnik. Priznanje za dosežek leta pripada Benjaminu Bajdu za zlato medaljo na računalniški olimpijadi. »Informacijsko limono« za najmanj primerno informacijsko tematiko je prejela nekompatibilnost zdravstvenih sistemov v Sloveniji, »informacijsko jagodo« kot najboljšo potezo pa dobi ekipa RTV za portal dostopno.si. Čestitke nagrajencem!

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

FOREWORD - INFORMATION SOCIETY 2023

The twenty-sixth Information Society multi-conference is taking place during a period of exceptional development for artificial intelligence, computing, and informatics, encompassing the entire information society. Generative artificial intelligence has made significant progress towards superintelligence, towards singularity, and the flourishing of human civilization with programs like ChatGPT. Experts' predictions are coming true, asserting that the mentioned fields are crucial for humanity's existence and development. Hence, we must direct our attention to them, swiftly integrating them into primary, secondary education, and the daily lives of individuals and communities.

On the other hand, alongside fake news, we witness the emergence of false encyclopaedias, pseudo-sciences, and flat Earth theories, along with the continuing neglect of scientific insights and methods, the diminishing of human rights, and societal values. It is upon all of us to appropriately address today's challenges, mainly assisting in the introduction of scientific knowledge and clearing up misconceptions. A frequently mentioned concern over the past year is the existential threat posed by artificial intelligence, supposedly endangering humanity as nuclear wars do. Yet, nobody provides a reasonably coherent scenario of how this might happen, say, how a 100x smarter GPT could endanger people.

This year's conference, besides purely technological aspects, highlights important integral themes like the environment, healthcare, depopulation policies, and solutions brought by artificial intelligence to almost all problems. In such an environment, in-depth analysis and discourse are crucial, shaping the best approaches to managing and exploiting technologies. We are fortunate to host a series of exceptional thinkers, scientists, and experts who bring a wealth of knowledge and dialogue in a collaborative and academically open environment. We believe their presence and participation are key to shaping a more inclusive, safe, and sustainable information society. For flourishing.

This year, we connected ten excellent independent conferences into the multi-conference, including "Legends of Computing", which introduces a new mechanism for promoting the information society. IS 2023 encompasses around 160 presentations, abstracts, and papers within standalone conferences and workshops. In total about 500 participants attended the conference. The event was accompanied by panel discussions, debates, and special events like the award ceremony. Selected contributions will also be published in a special issue of the journal Informatica (http://www.informatica.si/), boasting a 46-year tradition of being an excellent scientific journal. The Information Society 2023 multi-conference consists of the following independent conferences:

- Data Mining and Data Warehouse - SIKDD
- Demographic and Family Analysis
- Legends of Computing and Informatics
- Healthy Longevity Conference
- Myths and Truths about Environmental Protection
- International Conference on Technology Transfer
- Digital Inclusion in the Information Society - DIGIN 2023
- Slovenian Conference on Artificial Intelligence + DATASCIENCE
- Cognitive Science
- Education and Training in the Information Society
- Closing Conference Ceremony

Co-organizers and supporters of the conference include various research institutions and associations, among them ACM Slovenia, SLAIS for Artificial Intelligence, DKZ for Cognitive Science, and the Engineering Academy of Slovenia (IAS). On behalf of the conference organizers, we thank the associations and institutions, and especially the participants for their valuable contributions and the opportunity to share their experiences about the information society with us. We also thank the reviewers for their assistance in reviewing.

With the awarding of prizes, especially the Michie-Turing Award, the autonomous profession from the field identifies the most outstanding achievements. Prof. Dr. Andrej Brodnik received the Michie-Turing Award for his exceptional lifetime contribution to the development and promotion of the information society. The Achievement of the Year award goes to Benjamin Bajd, gold medal winner at the Computer Olympiad. The "Information Lemon" for the least appropriate information move was awarded to the incompatibility of information systems in the Slovenian healthcare, while the "Information Strawberry" for the best move goes to the RTV SLO team for portal dostopno.si. Congratulations to the winners!

Mojca Ciglarič, Chair of the Program Committee
Matjaž Gams, Chair of the Organizing Committee

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
Mateja Mavrič

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 26. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2023

Zvezek C

Proceedings of the 26th International Multiconference

INFORMATION SOCIETY – IS 2023

Volume C

Odkrivanje znanja in podatkovna skladišča – SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

9. oktober 2023 / 9 October 2023
Ljubljana, Slovenia

PREDGOVOR

Tehnologije, ki se ukvarjajo s podatki so v devetdesetih letih močno napredovale. Iz prve faze, kjer je šlo predvsem za shranjevanje podatkov in kako do njih učinkovito dostopati, se je razvila industrija za izdelavo orodij za delo s podatkovnimi bazami, prišlo je do standardizacije procesov, povpraševalnih jezikov itd. Ko shranjevanje podatkov ni bil več poseben problem, se je pojavila potreba po bolj urejenih podatkovnih bazah, ki bi služile ne le transakcijskem procesiranju ampak tudi analitskim vpogledom v podatke – pojavilo se je t.i. skladiščenje podatkov (data warehousing), ki je postalo standarden del informacijskih sistemov v podjetjih. Paradigma OLAP (On-Line-Analytical-Processing) zahteva od uporabnika, da še vedno sam postavlja sistemu vprašanja in dobiva nanje odgovore in na vizualen način preverja in išče izstopajoče situacije. Ker seveda to ni vedno mogoče, se je pojavila potreba po avtomatski analizi podatkov oz. z drugimi besedami to, da sistem sam pove, kaj bi utegnilo biti zanimivo za uporabnika – to prinašajo tehnike odkrivanja znanja v podatkih (data mining), ki iz obstoječih podatkov skušajo pridobiti novo znanje in tako uporabniku nudijo novo razumevanje dogajanj zajetih v podatkih. Slovenska KDD konferenca pokriva vsebine, ki se ukvarjajo z analizo podatkov in odkrivanjem znanja v podatkih: pristope, orodja, probleme in rešitve.

Dunja Mladenić
Marko Grobelnik

FOREWORD

Data driven technologies have significantly progressed after mid 90's. The first phases were mainly focused on storing and efficiently accessing the data, resulted in the development of industry tools for managing large databases, related standards, supporting querying languages, etc. After the initial period, when the data storage was not a primary problem anymore, the development progressed towards analytical functionalities on how to extract added value from the data; i.e., databases started supporting not only transactions but also analytical processing of the data. At this point, data warehousing with On-Line-Analytical-Processing entered as a usual part of a company's information system portfolio, requiring from the user to set well defined questions about the aggregated views to the data. Data Mining is a technology developed after year 2000, offering automatic data analysis trying to obtain new discoveries from the existing data and enabling a user new insights in the data. In this respect, the Slovenian KDD conference (SiKDD) covers a broad area including Statistical Data Analysis, Data, Text and Multimedia Mining, Semantic Technologies, Link Detection and Link Analysis, Social Network Analysis, Data Warehouses.

Dunja Mladenić
Marko Grobelnik

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Janez Brank, Jožef Stefan Institute, Ljubljana

Marko Grobelnik, Jožef Stefan Institute, Ljubljana

Branko Kavšek, University of Primorska, Koper

Besher M. Massri, Jožef Stefan Institute, Ljubljana

Dunja Mladenić, Jožef Stefan Institute, Ljubljana

Erik Novak, Jožef Stefan Institute, Ljubljana

Inna Novalija, Jožef Stefan Institute, Ljubljana

Jože Rožanec, Qlector, Ljubljana

Abdul Sitar, Jožef Stefan Institute, Ljubljana

Luka Stopar, Sportradar, Ljubljana

Jan Šturm, Jožef Stefan Institute, Ljubljana

Forecasting Trends in Technological Innovations with Distortion-Aware Convolutional Neural Networks

Krisztian Buza, M. Besher Massri, Marko Grobelnik
{krisztian.antal.buza,besher.massri,marko.grobelnik}@ijs.si
Artificial Intelligence Laboratory, Institute Jozef Stefan
Ljubljana, Slovenia

ABSTRACT

Predicting trends in technological innovations holds critical importance for policymakers, investors, and other stakeholders within the innovation ecosystem. This study approaches this challenge by framing it as a time series prediction task. Recent efforts have introduced diverse solutions utilizing convolutional neural networks, including distortion-aware convolutional neural networks. While convolutional layers act as local pattern detectors, conventional convolution matches local patterns in a rigid manner in the sense that they do not account for local shifts and elongations, whereas distortion-aware convolution incorporate the capability to identify local patterns with flexibility, accommodating local shifts and elongations. The resulting convolutional neural network, with distortion-aware convolution, has exhibited superior performance compared to standard convolutional networks in multiple time series prediction tasks. As a result, we advocate for the application of distortion-aware convolutional networks in forecasting technological innovation trends and compare their performance with conventional convolutional neural networks.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

trends, innovation ecosystem, time series forecasting, convolutional neural networks, distortion-aware convolution

1 INTRODUCTION

Forecasting trends in technological innovations is of high value for policy makers, investors and other actors of the innovation ecosystem. In this paper, we cast this task as a time series forecasting problem.

Approaches for time series forecasting range from the well-known autoregressive models [4] over exponential smoothing [12] to solutions based on deep learning [10, 11, 16–19, 24, 26]. Among the numerous techniques, a prominent family of methods include forecast with convolutional neural networks (CNNs) [3, 20].

The inherent assumption behind CNNs is that local patterns are characteristic to time series and future values of the time series may be predicted based on those local patterns. While the operation of convolution plays the role of a local pattern detector, it matches patterns in a rigid manner as it does not allow for local shifts and elongations within the patterns. This issue has been addressed by distortion-aware convolution and the resulting convolutional neural network has been shown to outperform conventional convolutional networks in case of several time series forecasting tasks [6].

For the aforementioned reasons, in this paper we propose to use distortion-aware convolutional networks for forecasting trends in technological innovations. We perform experiments on real-world time series of the number of patents related to selected topics. We compare the performance of distortion-aware convolutional networks with conventional convolutional neural networks.

The reminder of the paper is organized as follows. In Section 2, we provide a short discussion of related works. We review distortion-aware convolutional networks in Section 3, followed by the experimental results in Section 4. Finally, we conclude in Section 5.

2 RELATED WORK

As we cast our problem as a time series forecasting task, we focus our review of related works on time series forecasting. As mentioned previously, a prominent family of methods include forecast techniques based on convolutional neural networks, recent surveys about them have been presented by Lim et al. [17], Sezer et al. [21] and Torres et al. [24].

An essential component of distortion-aware convolution is dynamic time warping (DTW). While DTW is one of the most successful distance measures in the time series domain, see e.g. [25], recent approaches integrate it with neural networks. For example, Iwana et al. [14], Cai et al. [9] and Buza [5] used DTW to construct features. In contrast, Afrasiabi et al. [1] used neural networks to extract features and used DTW to compare the resulting sequences. Shulman [22] proposed "an approach similar to DTW" to allow for flexible matching in case of the dot product. DTW-NN [13] considered neural networks and replaced "the standard inner product of a node with DTW as a kernel-like method". However, DTW-NN only considered multilayer perceptrons (MLP), whereas we focus on convolutional networks. In the context of time series classification, Buza and Antal proposed to replace the dot product in the convolution operation by DTW calculations [7]. In distortion-aware convolution [6], DTW is used together with the dot product, but the dot product itself is not modified.

3 BACKGROUND

We begin this section with a formal definition of our task followed by a review of convolutional neural networks with distortion-aware convolution [6].

Slovenian KDD Conference 2023 , 9–13 October 2023, Ljubljana, Slovenia

3.1 Problem Formulation

Given an observed time series $x = (x_1, \ldots, x_l)$ of length l, in our case each x_i represents the number of patents related to a given topic in a month, we aim at predicting its subsequent h values $y = (x_{l+1}, \ldots, x_{l+h})$, i.e., the number of patents in the subsequent h months. We say that h is the forecast horizon and y is the target. Furthermore, we assume that a dataset D is given which contains n time series with the corresponding target:

$$D = \{(x^{(i)}, y^{(i)})_{i=1}^{n}\}. \quad (1)$$

We use D to train neural networks for the aforementioned prediction task. We say that $x^{(i)}$ is the input of the neural network.

In our experiments, we assume that an independent dataset D^* is given which can be used to evaluate the predictions of our model. Similarly to D, dataset D^* contains pairs of input and target time series. D^* is called the test set.

3.2 The Distortion-aware Convolutional Block

The main idea behind distortion-aware convolution [6] is to calculate, besides the dot products (or inner products), DTW distances between the kernel and time series segments as well. This is illustrated in Fig. 1. Our distortion-aware convolutional block has two output channels: one for dot products and another channel for the DTW distances.

While in case of the dot product, higher similarity between the time series segment and the pattern corresponds to higher values, the opposite is true for the DTW distances. In case of DTW, high similarity between the time series segment and the pattern is reflected by a distance close to zero. Therefore, to make sure that the activations on both channels are consistent, the activations of the DTW channel of our distortion-aware convolutional block are calculated as follows:

$$out_{DTW}(t) = \frac{1}{1 + DTW(in[t : t+s], w)}, \quad (2)$$

where out_{DTW} denotes the activation of the DTW channel of the distortion-aware convolutional block, $in[t : t+s]$ is the segment of the block's input between the t-th and $(t+s)$-th position[1], s is the size of the filter, w are the weights of the filter representing a local pattern and $DTW(.,.)$ is a function that calculates the DTW distance between two time series segments.

Training neural networks with distortion-aware convolution may be challenging because of the backpropagation of gradients through the DTW calculations. The basic idea of training is to train the network with conventional convolution instead of distortion-aware convolution initially and add DTW-computations once the weights of the convolutional layer have already been determined. For details, see [6].

4 EXPERIMENTAL EVALUATION

The goal of our experiments is to examine whether the neural networks with distortion-aware convolution are more suitable for forecasting technological trends compared to their counterparts with conventional convolution.

[1]In Eq. (2) we use a Python-like syntax: the lower index, t is inclusive, the upper index, $t+s$ is exclusive in $in[t : t+s]$.

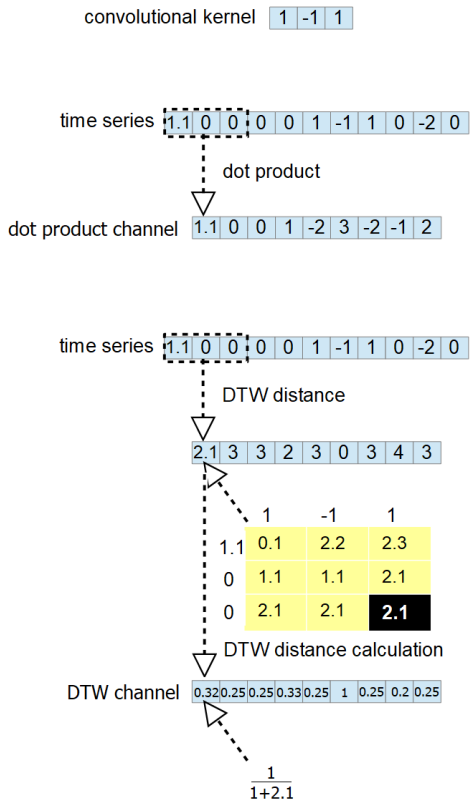


Figure 1: In case of distortion-aware convolution, additionally to the dot product (top), DTW distances between the kernel and time series segments are calculated (bottom). Thus, our distortion-aware convolutional block has two output channels: one for dot products and another channel for the DTW distances scaled according to Eq. (2).

4.1 Data

Lens is a web-based service that offers global access to patent information, academic articles, regulatory databases, and additional relevant materials.[2] The platform is designed to simplify the exploration and evaluation of intellectual property information while promoting research and inventive activities. Lens grants complimentary access to patent databases from more than 100 nations and includes sophisticated search functionalities and analytical tools for diverse research and analysis needs.

We extracted time series from the Lens patent database as follows. For selected topics identified by their Cooperative Patent Classification (CPC) codes, we extracted the number of granted patents as well as the number of patent applications per month between January 1980 and December 2022. We considered the following topics: (a) "image or video recognition" (G06V), (b) "neural networks" (G06N3/02), (c) "natural language processing" (G06F40) and (d) all topics related to artificial intelligence. We considered the number of patents separately for the most significant jurisdictions, i.e., (a) United States of America, (b) China, (c) Korea, (d) Japan and

[2]http://lens.org

Table 1: Mean absolute error (MAE) and root mean squared error (RMSE) for forecasting the time series of granted patents in case of our approach (DCNN) and the baseline (CNN). Lower values indicate better performance.

topic	jurisdiction	RMSE CNN	RMSE DCNN	MAE CNN	MAE DCNN
image or video recognition	US	165.9	106.0	131.2	92.7
	China	405.8	320.9	323.87	217.6
	Korea	13.9	27.7	12.4	19.9
	Japan	55.9	49.8	39.9	37.8
	Europe	34.5	34.7	32.3	32.9
	ALL	494.7	399.6	416.8	341.3
neural networks	US	10.7	9.1	9.4	7.9
	China	5.6	5.5	3.8	3.7
	Korea	6.3	2.3	5.4	2.1
	Japan	3.5	2.9	2.5	2.0
	Europe	2.7	1.6	2.2	1.2
	ALL	7.6	8.3	6.3	6.7
natural language processing	US	19.7	15.1	14.8	12.0
	China	57.1	47.0	41.6	41.7
	Korea	14.2	8.5	13.1	7.3
	Japan	11.8	10.7	9.5	7.3
	Europe	4.8	3.0	3.5	2.7
	ALL	67.0	45.7	59.5	35.5
ALL	US	270.2	216.9	224.1	196.4
	China	870.2	1108.8	763.2	998.1
	Korea	56.6	138.3	53.8	129.4
	Japan	124.8	132.0	81.4	89.9
	Europe	85.8	69.2	82.1	65.9
	ALL	1045.1	1129.1	929.2	964.6

Table 2: Mean absolute error (MAE) and root mean squared error (RMSE) for forecasting the time series of patent applications in case of our approach (DCNN) and the baseline (CNN). Lower values indicate better performance.

topic	jurisdiction	RMSE CNN	RMSE DCNN	MAE CNN	MAE DCNN
image or video recognition	US	188.2	177.1	170.2	163.3
	China	3405.0	1061.7	3375.4	1042.3
	Korea	128.9	70.8	99.7	69.4
	Japan	103.8	106.4	87.1	66.1
	Europe	51.9	55.5	45.0	49.4
	ALL	3641.9	2110.5	3627.3	2027.8
neural networks	xUS	79.8	15.3	76.9	12.7
	China	21.2	20.8	16.8	19.0
	Korea	44.6	6.8	43.7	6.2
	Japan	13.9	7.1	13.5	4.8
	Europe	15.8	5.9	14.9	4.4
	ALL	267.7	45.6	262.7	38.6
natural language processing	US	64.1	68.7	55.5	64.6
	China	418.9	318.2	363.6	289.3
	Korea	35.1	23.4	29.7	21.0
	Japan	16.7	18.7	10.5	10.8
	Europe	11.2	14.3	9.7	11.2
	ALL	298.1	543.0	226.9	489.3
ALL	US	532.3	329.1	458.9	311.3
	China	6443.7	2784.2	6239.0	2386.5
	Korea	405.4	216.8	340.2	180.8
	Japan	224.8	228.1	159.1	128.6
	Europe	130.0	163.5	97.5	121.3
	ALL	5445.1	3355.8	5009.0	2547.0

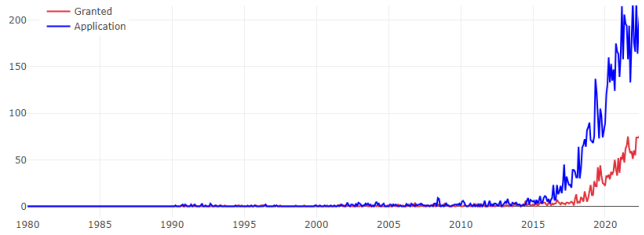


Figure 2: Total number of granted patents (red) and patent applications (blue) for all the jurisdictions in the Lens database related to "neural networks" (CPC: G06N3/02).

(e) Europe. Additionally, we considered the time series of the total number of patents for all the jurisdictions of the database. Thus, we considered in 48 time series in total, see also the first two columns of Tab. 1 and Tab. 2. Two example time series are shown in Fig. 2.

For each time series, we trained the neural networks to predict the number of granted patents (or patent applications, respectively) for each month of a 6-monthly period, i.e., the forecast horizon was $h = 6$. As input, we used the number of granted patents (or patent applications, respectively) in the previous 36 months. The data related to the years 1980...2019 was used as training data, while the data from 2019...2022 was used as test data.

From the long time series corresponding years 1980...2019, we extracted training instances with a moving window. This resulted in 10496 training instances in total which corresponds to 427 training instance for each time series.

When evaluating the network on the test data, we used the data from 2019...2021 as input data and the task was to predict the number of granted patents (or patent applications, respectively) for the first six month of 2022.

4.2 Experimental Settings

In order to assess the contribution of distortion-aware convolution, for each time series, we trained two versions of the neural network: *with* and *without* distortion-aware convolution, and compared the results. In the former case, the first hidden layer was a distortion-aware convolutional layer (with both dot product and DTW calculations), whereas in the later case, we used conventional convolution (with dot product only).

For simplicity, we considered a convolutional network containing a single convolutional layer with 25 filters, followed by a max pooling layer with window size of 2, and a fully connected layer with 100 units. We set the size of convolutional filters to 9. The

number of units in the output layer corresponds to the forecast horizon, as each unit is expected to predict one of the numeric values of the target time series. We trained the networks for 1000 epochs with the Adam optimizer [15] with learning rate of 10^{-5} and batch size of 16. The loss function was mean squared error.

We implemented our neural networks in Python using the PyTorch framework. In order to support reproduction of our work, we made the implementation of our model publicly available in a github repository. The code illustrates training and evaluation of our model on standard benchmark datasets.[3]

We evaluated the predicted time series both in terms of mean absolute error (MAE) and root mean squared error (RMSE). In particular, we calculated MAE (and RMSE, respectively) for each forecast time series.

As the goal of our experiments is to assess the contribution of distortion-aware convolution, our baseline, denoted as CNN, is the aforementioned neural network with conventional convolution instead of distortion-aware convolution.

4.3 Results

Tab. 1 and Tab. 2 show our results in terms of MAE and RMSE. Our approach, convolutional neural network with distortion-aware convolution is denoted by DCNN, while CNN denotes the neural network with conventional convolution. As one can see, in the majority of the examined cases, DCNN outperforms CNN both in terms of MAE and RMSE. In those cases when CNN performs better, typically, both models are rather accurate (the error is low for both models) or the difference is very small compared to the magnitude of the error.

5 CONCLUSIONS AND OUTLOOK

In this paper, we focused on forecasting technological trends and cast this task as a time series forecasting problem. We considered a recent approach, convolutional neural networks with distortion-aware convolution, which has not been used for this task previously.

We performed experiments on real-world time series representing the number of granted patents and patent applications related to selected topics. Our observations show that convolutional neural networks with distortion-aware convolution are promising for this task. Furthermore, combination of conventional convolutional networks and neural networks with distortion-aware convolution may be an interesting target of future works.

Last, but not least, we mention that time series are prominent in various real-world applications [2, 23] and our approach can be extended to handle other types of time series, such as multivariate time series (or series of vectors) that can be compared with a more general version of DTW, see e.g. [8].

ACKNOWLEDGMENTS

This work was supported by the European Union through enRichMyData EU HE project under grant agreement No 101070284.

[3] https://github.com/kr7/dcnn-forecast

REFERENCES

[1] Mahlagha Afrasiabi, Muharram Mansoorizadeh, et al. 2019. DTW-CNN: time series-based human interaction prediction in videos using CNN-extracted features. *The Visual Computer* (2019), 1–13.
[2] Margit Antal and László Zsolt Szabó. 2016. On-line verification of finger drawn signatures. In *11th international symposium on applied computational intelligence and informatics*. IEEE, 419–424.
[3] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. 2017. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691* (2017).
[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.
[5] Krisztian Buza. 2020. Asterics: Projection-based classification of eeg with asymmetric loss linear regression and genetic algorithm. In *14th International Symposium on Applied Computational Intelligence and Informatics*. IEEE, 000035–000040.
[6] Krisztian Buza. 2023. Time Series Forecasting with Distortion-Aware Convolutional Neural Networks. In *9th SIGKDD International Workshop on Mining and Learning from Time Series*.
[7] Krisztian Buza and Margit Antal. 2021. Convolutional neural networks with dynamic convolution for time series classification. In *International Conference on Computational Collective Intelligence*. Springer, 304–312.
[8] Krisztian Antal Buza. 2011. Fusion methods for time-series classification. *PhD thesis at the University of Hildesheim* (2011).
[9] Xingyu Cai, Tingyang Xu, Jinfeng Yi, Junzhou Huang, and Sanguthevar Rajasekaran. 2019. DTWNet: a dynamic time warping network. *Advances in neural information processing systems* 32 (2019).
[10] Zhengping Che, Sanjay Purushotham, Guangyu Li, Bo Jiang, and Yan Liu. 2018. Hierarchical deep generative models for multi-rate multivariate time series. In *International Conference on Machine Learning*. PMLR, 784–793.
[11] Marco Cuturi and Mathieu Blondel. 2017. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*. PMLR, 894–903.
[12] Everette S Gardner Jr. 2006. Exponential smoothing: The state of the art—Part II. *International journal of forecasting* 22, 4 (2006), 637–666.
[13] Brian Kenji Iwana, Volkmar Frinken, and Seiichi Uchida. 2020. DTW-NN: A novel neural network for time series recognition using dynamic alignment between inputs and weights. *Knowledge-Based Systems* 188 (2020), 104971.
[14] Brian Kenji Iwana and Seiichi Uchida. 2020. Time series classification using local distance-based features in multi-modal fusion networks. *Pattern Recognition* 97 (2020), 107024.
[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[16] Vincent Le Guen and Nicolas Thome. 2019. Shape and time distortion loss for training deep time series forecasting models. *Advances in neural information processing systems* 32 (2019).
[17] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200209.
[18] Linbo Liu, Youngsuk Park, Trong Nghia Hoang, Hilaf Hasson, and Luke Huan. 2023. Robust Multivariate Time-Series Forecasting: Adversarial Attacks and Defense Mechanisms. In *The Eleventh International Conference on Learning Representations*.
[19] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).
[20] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. 2019. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems* 32 (2019).
[21] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing* 90 (2020), 106181.
[22] Yaniv Shulman. 2019. Dynamic Time Warp Convolutional Networks. *arXiv preprint arXiv:1911.01944* (2019).
[23] Abdul Sittar and Dunja Mladenić. 2023. An approach to creating a time-series dataset for news propagation: Ukraine-war case study. In *Slovenian KDD Conference*.
[24] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. 2021. Deep learning for time series forecasting: a survey. *Big Data* 9, 1 (2021), 3–21.
[25] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. 2006. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*. 1033–1040.
[26] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. 2022. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems* 35 (2022), 12677–12690.

Building A Causality Graph For Strategic Foresight

Jože M. Rožanec
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
joze.rozanec@ijs.si

Beno Šircelj
Jožef Stefan Institute
Ljubljana, Slovenia
beno.sircelj@ijs.si

Peter Nemec
Event Registry d.o.o.
Ljubljana, Slovenia
peter@eventregistry.org

Gregor Leban
Event Registry d.o.o.
Ljubljana, Slovenia
gregor@eventregistry.org

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

This paper describes a pipeline built to generate a causality graph for strategic foresight. The pipeline interfaces with a well-known global media retrieval platform, which performs real-time tracking of events reported in the media. The events are retrieved from the media retrieval platform, and content from the media articles is processed with ChatGPT to extract causal relations mentioned in the news article. Multiple post-processing steps are performed to clean the causal relations, removing spurious ones and linking them to ontological concepts where possible. Finally, a sample causality trace is showcased to exemplify the potential of the causality graph created so far.

KEYWORDS

strategic foresight, graph, causality extraction, wikifier, ChatGPT

1 INTRODUCTION

Among the most frequently used strategic foresight methods we find scenario planning [7], that aims to foresee relevant scenarios based on trends and factors of influence. These allow for a better understanding of how actions can influence the future - a key ability in a world full of Turbulence, Unpredictability, Uncertainty, Novelty, and Ambiguity (TUNA) [30]. This ability has fostered an increasing adoption of strategic foresight in the public and private sectors [6, 21].

Domain experts currently plan scenarios by gathering and analyzing the data to determine and report probable, possible, and plausible futures of interest [15]. Nevertheless, the extensive manual work imposes severe scalability limitations and can introduce bias into the assessments [7]. To overcome such limitations, artificial intelligence was proposed to automate information scanning and data analysis [4, 18].

While the value of artificial intelligence for strategic foresight has been recognized, artificial intelligence has not been widely adopted yet [4, 20]. This is also reflected in scientific papers on foresight and artificial intelligence. For example, we queried Google Scholar for "data-supported foresight" and "strategic foresight artificial intelligence" considering the start time is unlimited, and the deadline is September 6th 2023. When analyzing the first 50 search results of each, we got 18% (9/50) and 40% (20/50) relevant hits, respectively. Some approaches described in the literature aim to leverage artificial intelligence to automate time-consuming aspects of strategic foresight, such as performing information scanning and data analysis [4, 18]. Furthermore, text-mining techniques have been used to identify weak signals and trends [10] or extract relevant actions and outcomes that could be mapped to causal decision diagrams [19].

Strategic foresight for environmental purposes has been considered to different degrees by countries and environmental agencies. For example, multiple U.S. Environmental Protection Agency offices began using strategic foresight in the 1980s. Still, they did not do so consistently until 1995, when it began to be institutionalized and connected to the Agency's strategic planning and decision-making, and reinvigorated since 2015 with that purpose [11]. Another example is The Netherlands, where strategic foresight has been encouraged since 1992 to systematically aim to identify critical technologies and scientific possibilities that would allow the fulfillment of environmental policies [29]. Other cases include using strategic foresight to understand how EU-wide policies may affect regions and rural localities [26] or guide decision-making in the face of structural change [2].

Previous work [22, 23] described how artificial intelligence could be used to automate scenario planning. This paper describes a pipeline built to extract and process media news from EventRegistry [16] to create a causality graph. Furthermore, it describes the causality graph created with media news reporting on events related to oil prices, given the abundant research regarding how oil prices impact the environment. Among the benefits of this approach is the ability to extract causal relations with little human intervention and no supervision. The resulting graph enables the creation of link prediction models that can be used to predict future events based on an array of events that have been observed in the past.

This paper is organized as follows. First, section 2 describes how a data extraction pipeline was built, retrieving media events of interest and extracting causal relationships observed in the world and described in them. Section 3 briefly describes some of the results obtained, providing (i) a quantitative assessment of error types and resulting causal relationships after data cleansing procedures and (ii) a qualitative assessment of causality relationships generated through the pipeline. Finally, Section 4 concludes and outlines future work.

2 DATA EXTRACTION PIPELINE

The data extraction pipeline aims to query relevant media news, process them, and extract causal relationships that can be modeled in a graph. Given the specific interest in modeling causality

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

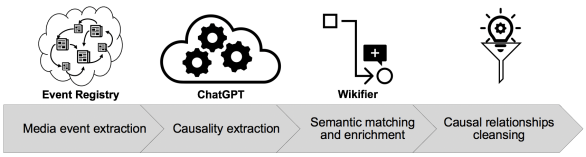


Figure 1: Data extraction pipeline used to retrieve media events and extract causal relationships.

for environmental protection, some research was performed to identify possible topics of interest. Among potential topics, the influence of oil prices on the environment was selected, considering such a topic is frequently covered in the media and was researched to a certain extent. Research has shown that oil price fluctuations (a) affect the consumption of renewable energy sources [1, 28], (b) stimulate green innovation, and that positive shocks in oil prices reduce CO_2 emissions [12], and enhance ecological quality [8, 14].

The data extraction pipeline is summarized in Fig. 1, and each component is briefly described in the following subsections.

2.1 Media Event Extraction

The EventRegistry platform provides real-time insights into media events by sourcing them from the News Feed service [27], processing them and creating media events based on cross-lingual clusters of media news, which are later exposed through an API. The news processing steps require news semantic annotation, extraction of date references, cross-lingual matching, and detection of news duplicates. The cross-lingual clusters denoting a particular media event have a summary describing the media event, information regarding the piece of news considered a centroid to the cluster, and other relevant information.

The first step in the pipeline queries the EventRegistry media event API to extract media events related to a particular concept. This research's query concept was limited to the "Price of Oil". Since EventRegistry has a history of data up to 2014, relevant geopolitical and economic events that influenced oil prices since 2014 were searched. Two events were highlighted by the U.S. Energy Information Administration [1]: (a) the fact that OPEC production quota remained unchanged in the first quarter of 2015 and (b) a reduction in oil demand registered due to the global pandemic in the first quarter of 2020. Furthermore, events between 2022 and 2023 were considered, given the impact of the Russo-Ukrainian War on oil prices [17]. For each event obtained, the centroid media news was queried, its text extracted, wikified, and stored for further processing.

2.2 Causality extraction

To extract causal relations from media events, the OpenAI ChatGPT (gpt-3.5-turbo) was used as a one-shot learning model. To that end, a random media event was sampled, the causality relationships extracted, and both (the text and causal relationships) presented to the model, asking it to recognize causal relationships in the media news. Several iterations of prompt engineering were performed to ensure high-quality results, performing a manual assessment of random results.

The causal relationships persisted in JSON files discriminated the cause, effect, related entities, and locations. In particular, cause, effect, entities, and locations were defined in the following manner:

- **Cause or effect**: contains an entity which is an item, individual, or company that an event happened to;
- **Event**: is an action, development, happening, or state of the entity that is causing or was affected by a cause in the relationship;
- **Location**: geographical location where the event in the cause or effect took place;

Once the causal relationships were extracted, the cause and effect were post-processed, removing adjectives so that only the nouns were left. E.g., *higher diesel prices* was converted to *diesel prices*. The decision was made considering that by doing so, (a) the causes and effects would gain greater support and, therefore, strengthen the information signal in a graph, and (b) that a human expert would be able to determine how a cause and effect may relate given his domain knowledge and a particular context. For example, given the relationship *Inflation → Consumer price index*, the human expert will immediately understand how the consumer price index is affected in a growing or shrinking inflationary context. For each causal relationship, a trace was kept to associate them with the media event from which they were extracted to enable further analysis when required.

2.3 Semantic matching and enrichment

The entire text of the media article was parsed using Wikifier [5]. Data from Wikifier was employed in two distinct ways: firstly, to enrich location data, and secondly, to associate entities to relevant semantic concepts.

The Wikifier tool marks which words in the wikified text correspond to certain semantic concepts. Such annotations were matched to the entities extracted by ChatGPT as part of the causal relationships. To successfully match strings to semantic concepts, some preprocessing was required. First, the non-letter symbols and stopwords were removed, followed by the stemming of each word. It was considered a match if at least one identical string between the text related to marked concepts and the causal relationship. Not all of the semantic concepts listed by the Wikifier were considered: (a) the concepts were required to have a PageRank higher than 0.0001; (b) for location data, only the concepts categorized as *"place"* were considered, and (c) when substituting the original entity by the associated semantic concept, the semantic concept with the highest cosine similarity between the article it's corresponding Wikipedia page was considered.

2.4 Cleansing causal relations

After extracting causal relations, we focused on analyzing the data and cleansing to ensure only relevant relations were considered and used to build a causality graph. Subsequent random sampling iterations were performed, extracting 300 causal relationships in each iteration, which were then analyzed. In each iteration, the causal relations were assessed to determine whether they were meaningful to the topic under consideration, to identify common errors, and to propose mitigation strategies that could amend such errors or filter useless causal relations. We typified six such cases, five originating from ChatGPT and one when semantically post-processing the causal relations with concepts obtained from the Wikifier:

- **repeated entity**: [ChatGPT] the same entity is registered for cause and effect. E.g., *Oil price → Oil price*.

[1]The events were highlighted in the following report, last accessed on August 25th 2023: https://www.eia.gov/finance/markets/crudeoil/spot_prices.php.

- **empty entity**: [ChatGPT] an entity is missing as cause or effect. E.g., → *Oil price*.
- **missing entity**: [ChatGPT] ChatGPT omits the actual entity but could be inferred from the text by the human reader. E.g., *S&P 500 capital expenditures* → *growth*, *energy policy* → *defiance*, or *survey* → *Nasdaq 100*.
- **time entity**: [ChatGPT] some time-period is considered an entity. E.g., *drilling activity* → *2016*, or *(US) shale oil supply* → *end of the year*.
- **non-entity**: [ChatGPT] words marked as entities don't mean anything coherent. E.g., *retail sales* → *risk appetite*.
- **wrong conversion**: [Wikifier] the entity was changed to something unrelated to the one stated in the text. E.g., *Australian government* > *Australian dollar*, or *political tensions* > *Breakup of Yugoslavia*.

While the mitigation strategy for most of the abovementioned errors is to remove the causal relationship, for *missing entity*, a follow-up question will be provided to ChatGPT to get a more concrete answer. This last mitigation strategy has not been implemented yet. Furthermore, a list of concept mappings will be considered to reduce clutter. For example, *Wage Growth* or *1980s Oil Glut* should be replaced by *Wage* or *Oil Glut*, respectively. *Breakup of Yugoslavia* could be replaced by *Country Breakup*. Finally, a more thorough linking to semantic concepts and ontologies is required (e.g., *Jerome Powell* could be linked to *Central Bank*).

After the abovementioned cleansing, the strings were turned into lowercase and trimmed, and most non-alphabetical characters were removed. Further sampling and entity evaluation were performed, creating a dictionary to match string occurrences to a particular concept. It must be noted that the dictionaries do not provide an exhaustive mapping and that ongoing work is being done to further refine and complete the mapping phase. Such dictionaries were created to provide ground for future ontological mapping based on existing ontologies and ontologies that will be developed for this purpose. Finally, all the relations that, after the described process, were extracted from only one media event were discarded, given they are very likely to introduce noise.

2.5 Creating a causality graph

Once causal relationships were extracted, a causality graph was created by matching $cause \rightarrow effect$. Furthermore, some metrics were computed to assess the graph characteristics. The graph can be sampled and visualized with the NetworkX [2] library, which creates a dynamic HTML interface to view it. For each cause and all the possible effects following it, probabilities of each effect occurring were computed based on the ratios present in the data.

3 RESULTS

A total of 2,503 media events were extracted from EventRegistry. When processed with ChatGPT, 12,290 unique causal relationships were extracted, totaling 14,226 unique entities. Those were processed to remove possible errors. Considering *repeated entity* and *empty entity* errors, 253 causal relations were removed. After applying wikification, 9,726 unique causal relations remained, totaling 7,723 entities. 845 causal relations were removed, considering *repeated entity* and *empty entity* errors. Table 1 shows the number of causal relations affected by a particular error type, considering a random sample of 300 causal relations.

[2]The library is documented at the following website: https://networkx.org/

Error type	Count	Percentage
Wrong conversion	17	5.7%
Missing entity	15	5.0%
Non-entity	9	3.0%
Time entity	3	1.0%

Table 1: Statistics for typified errors based on a random sample of 300 causal relationships.

After performing the abovementioned cleansing and dictionary-based mappings, 7,723 nodes and 9,726 edges were obtained. Removing causal relationships reported only in a single media event reduced the graph size to 489 nodes and 877 edges.

3.1 Causality graph and causality chain analysis

Causal chains were created by linking causes and effects extracted from media events. While these are not always completely accurate, they help to identify sequences of events that may take place. Furthermore, while currently not implemented, graph link prediction could be used to predict future event sequences based on patterns observed in the past.

This section provides an example regarding a causality chain of interest retrieved from the causality graph. The causality chain is briefly analyzed to demonstrate how it captures relevant knowledge. In particular, many causality chains displayed the following pattern: *Pandemic* → *Currency* → *Price of Oil* → *Economic Growth* → *Oil Glut* → *Inflation* → *Central Bank* → *Stock Market* → *Investment*.

The complete causality chain summarized above was: *Pandemic* → *Currency* → *Price of Oil* → *Crude Oil Futures* → *Fuel Pricing* → *Economic Growth* → *Petroleum* → *Oil Glut* → *Consumer Price Index* → *Monetary Policy* → *Inflation* → *Central Bank* → *Stock Market* → *Investment* → *Bond*.

To validate the causality chain, scientific literature and events from the past few years were reviewed to find research and examples to validate the causal relationships. For the causality chain described above, we found that the *Pandemic* influenced *Currency*: countries experiencing a sharp daily rise in COVID-19 deaths usually saw their currencies weaken [13]. Causality between exchange rates (*Currency*) and *Price of Oil* has been reported by the European Central Bank [9]. In particular, it has been noticed that the exchange rates can affect oil prices through financial markets, financial assets, portfolio rebalancing, and heading practices. It has also been noted that given the oil prices are expressed in US dollars, the oil futures can be used to hedge against an expected depreciation in US dollars - something that explains the causal relationship between *Price of Oil* and *Crude Oil Futures*. Furthermore, a relationship exists between futures and spot prices (futures prices tend to converge upon spot prices [3] and between oil prices and fuel prices[4], validating the causal relationship between *Crude Oil Futures* and *Fuel Pricing*.

[3]See *"Futures Prices Converge Upon Spot Prices"*, last accessed at https://www.investopedia.com/ask/answers/06/futuresconvergespot.asp in September 7th 2023.
[4]See *"Gasoline explained: Factors affecting gasoline prices"*, last accessed at https://www.eia.gov/energyexplained/gasoline/factors-affecting-gasoline-prices.php in September 7th 2023.

When considering the relationship *Fuel Pricing* and *Economic Growth*, we found that the relationship is validated with energy prices [3], e.g., with gas prices: higher gas prices negatively impact the economy[5]. Economic growth can affect the petroleum market and, in particular, lead to an oil glut (a significant surplus of crude oil caused by falling demand) as it happened at the beginning of the COVID-19 pandemic[6]. Furthermore, oil pricing can have direct or indirect effects on *Inflation* [24], which is reflected in the *Consumer Price Index*, and which can trigger a particular *Monetary Policy* from the *Central Bank* in response to it. Finally, monetary policies affect the stock market and investments [25].

While the causality chain displayed in this case is mostly clean, some improvements are required to make it neater. For example, based on domain knowledge, and depending on the context, the *Consumer Price Index* and *Inflation* could be merged into a single concept, and *Monetary Policy* and *Central Bank* could be considered as one.

The ingestion pipeline requires further work to enhance the concept mappings. We envision that the dictionaries will be further evolved and linked to specific ontologies that could be used to assign semantic meaning and, e.g., contract links in a chain with the same semantic ancestor.

4 CONCLUSIONS

This research has described a pipeline created for causality extraction from media news and aimed toward a strategic foresight tool, and currently focused on events affecting oil prices. Particular errors in the causality extraction were identified and typified, and mitigation measures were implemented. Nevertheless, further work is required to improve the pipeline. Future work will consider three directions: (a) string to ontologies mapping to ensure the captured causes and effects can be tied to particular semantic knowledge and exploit it, (b) generate richer cause and effect representations so that based on encoded metadata, better causality patterns can be elucidated, and (c) create a link prediction model based on the causality graph.

ACKNOWLEDGMENTS

The Slovenian Research Agency supported this work. This research was developed as part of the Graph-Massivizer project funded under the Horizon Europe research and innovation program of the European Union under grant agreement 101093202.

REFERENCES

[1] Nicholas Apergis and James E Payne. 2015. Renewable energy, output, carbon dioxide emissions, and oil prices: evidence from South America. *Energy Sources, Part B: Economics, Planning, and Policy* 10, 3 (2015), 281–287.
[2] M Bruce Beck. 2005. Environmental foresight and structural change. *Environmental Modelling & Software* 20, 6 (2005), 651–670.
[3] Istemi Berk and Hakan Yetkiner. 2014. Energy prices and economic growth in the long run: Theory and evidence. *Renewable and Sustainable Energy Reviews* 36 (2014), 228–235.
[4] Patrick Brandtner and Marius Mates. 2021. Artificial Intelligence in Strategic Foresight–Current Practices and Future Application Potentials: Current Practices and Future Application Potentials. In *The 2021 12th International Conference on E-business, Management and Economics*. 75–81.
[5] Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant wikipedia concepts. *Proceedings of SiKDD* 472 (2017).
[6] George Burt and Anup Karath Nair. 2020. Rigidities of imagination in scenario planning: Strategic foresight through 'Unlearning'. *Technological Forecasting and Social Change* 153 (2020), 119927.
[7] Ashkan Ebadi, Alain Auger, and Yvan Gauthier. 2022. Detecting emerging technologies and their evolution using deep learning and weak signal analysis. *Journal of Informetrics* 16, 4 (2022), 101344.
[8] Ali Ebaid, Hooi Hooi Lean, and Usama Al-Mulali. 2022. Do oil price shocks matter for environmental degradation? Evidence of the environmental Kuznets curve in GCC countries. *Frontiers in Environmental Science* 10 (2022), 860942.
[9] Marcel Fratzscher, Daniel Schneider, and Ine Van Robays. 2014. Oil prices, exchange rates and asset prices. (2014).
[10] Amber Geurts, Ralph Gutknecht, Philine Warnke, Arjen Goetheer, Elna Schirrmeister, Babette Bakker, and Svetlana Meissner. 2022. New perspectives for data-supported foresight: The hybrid AI-expert approach. *Futures & Foresight Science* 4, 1 (2022), e99.
[11] Joseph M Greenblott, Thomas O'Farrell, Robert Olson, and Beth Burchard. 2019. Strategic foresight in the federal government: a survey of methods, resources, and institutional arrangements. *World futures review* 11, 3 (2019), 245–266.
[12] Jinyan Hu, Kai-Hua Wang, Chi Wei Su, and Muhammad Umar. 2022. Oil price, green innovation and institutional pressure: A China's perspective. *Resources Policy* 78 (2022), 102788.
[13] Aamir Jamal and Mudaser Ahad Bhat. 2022. COVID-19 pandemic and the exchange rate movements: evidence from six major COVID-19 hot spots. *Future Business Journal* 8, 1 (2022), 17.
[14] Foday Joof, Ahmed Samour, Mumtaz Ali, Turgut Tursoy, Mohammad Haseeb, Md Emran Hossain, and Mustafa Kamal. 2023. Symmetric and asymmetric effects of gold, and oil price on environment: The role of clean energy in China. *Resources Policy* 81 (2023), 103443.
[15] Kevin Kohler. 2021. Strategic Foresight: Knowledge, Tools, and Methods for the Future. *CSS Risk and Resilience Reports* (2021).
[16] Gregor Leban, Blaz Fortuna, Janez Brank, and Marko Grobelnik. 2014. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web*. 107–110.
[17] Gaye-Del Lo, Isaac Marcelin, Théophile Bassène, and Babacar Sène. 2022. The Russo-Ukrainian war and financial markets: the role of dependence on Russian commodities. *Finance Research Letters* 50 (2022), 103194.
[18] Nathan H Parrish, Anna L Buczak, Jared T Zook, James P Howard, Brian J Ellison, and Benjamin D Baugher. 2019. Crystal cube: Multidisciplinary approach to disruptive events prediction. In *Advances in Human Factors, Business Management and Society: Proceedings of the AHFE 2018 International Conference on Human Factors, Business Management and Society, July 21-25, 2018, Loews Sapphire Falls Resort at Universal Studios, Orlando, Florida, USA 9*. Springer, 571–581.
[19] Lorien Pratt, Christophe Bisson, and Thierry Warin. 2023. Bringing advanced technology to strategic decision-making: The Decision Intelligence/Data Science (DI/DS) Integration framework. *Futures* 152 (2023), 103217.
[20] Norbert Reez. 2020. Foresight-Based Leadership. Decision-Making in a Growing AI Environment. In *International Security Management: New Solutions to Complexity*. Springer, 323–341.
[21] Aaron B Rosa, Niklas Gudowsky, and Petteri Repo. 2021. Sensemaking and lens-shaping: Identifying citizen contributions to foresight through comparative topic modelling. *Futures* 129 (2021), 102733.
[22] Joze Rozanec, Peter Nemec, Gregor Leban, and Marko Grobelnik. 2023. AI, What Does the Future Hold for Us? Automating Strategic Foresight. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*. 247–248.
[23] Jože M Rožanec, Radu Prodan, Dumitru Roman, Gregor Leban, and Marko Grobelnik. 2023. AI-based Strategic Foresight for Environment Protection. In *Symposium on AI, Data and Digitalization (SAIDD 2023)*. 7.
[24] Siok Kun Sek, Xue Qi Teo, and Yen Nee Wong. 2015. A comparative study on the effects of oil price changes on inflation. *Procedia Economics and Finance* 26 (2015), 630–636.
[25] Peter Sellin. 2001. Monetary policy and the stock market: theory and empirical evidence. *Journal of economic surveys* 15, 4 (2001), 491–541.
[26] Anastasia Stratigea and Maria Giaoutzi. 2012. Linking global to regional scenarios in foresight. *Futures* 44, 10 (2012), 847–859.
[27] Mitja Trampuš and Blaz Novak. 2012. Internals of an aggregated web news feed. In *Proceedings of 15th Multiconference on Information Society*. 221–224.
[28] Victor Troster, Muhammad Shahbaz, and Gazi Salah Uddin. 2018. Renewable energy, oil prices, and economic activity: A Granger-causality in quantiles analysis. *Energy Economics* 70 (2018), 440–452.
[29] Barend Van der Meulen. 1999. The impact of foresight on environmental science and technology policy in the Netherlands. *Futures* 31, 1 (1999), 7–23.
[30] Angela Wilkinson. 2017. Strategic foresight primer. *European Political Strategy Centre* (2017).

[5] See *"How Gas Prices Affect the Economy"*, last accessed at https://www.investopedia.com/financial-edge/0511/how-gas-prices-affect-the-economy.aspx in September 7th 2023.

[6] See *"Oil glut means there's little hope for oil price recovery until 2021"*, last accessed at https://www.conference-board.org/topics/natural-disasters-pandemics/COVID-19-oil-glut in August 30th 2023.

Towards Testing the Significance of Branching Points and Cycles in Mapper Graphs

Patrik Zajec
patrik.zajec@ijs.si
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Primož Škraba
p.skraba@qmul.ac.uk
School of Mathematical Sciences, Queen Mary University of London
London, UK

Dunja Mladenič
dunja.mladenic@ijs.si
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

Given a point cloud P, which is a set of points embedded in $\mathbb{R}^d$, we are interested in recovering its topological structure. Such a structure can be summarized in the form of a graph. An example of this is the mapper graph, which captures how the point cloud is connected and reflects the branching and cyclic structure of P as branching points (vertices with degree greater than 2) and cycles in the graph. However, such a representation is not always accurate, i.e., the structure shown by the graph may not be sufficiently supported in the point cloud. To this end, we propose an approach that uses persistent (relative) homology to detect branching and cyclic structure, and employs a statistical test to confirm whether the structure is indeed significant. We show how the approach works for low-dimensional point clouds, and discuss its possible applications to real world point clouds.

KEYWORDS

topological data analysis, statistical hypothesis testing, persistent homology, mapper algorithm

1 INTRODUCTION

Consider the point cloud P consisting of points in $\mathbb{R}^2$ shown in Figure 1a. Using the mapper algorithm, we can construct a graph that represents its topological structure like the one in Figure 1b, which seems to recover the important structure. Using the same algorithm (but with different values of its adjustable parameters) we could end up with different graphs. The second graph, shown in Figure 1c, contains two cycles: the middle one, which captures the cycle present in P, and the top one, where the algorithm "mistakenly" considers the top points to connect in a cycle. The third graph, shown in Figure 1d, shows a similar structure as the graph in Figure 1b, although it contains one branching point more (splitting off the upper left branch) and a cycle of length three. One could argue that these branching and cyclic structures are not sufficiently supported in P.

Our goal is to develop an approach that allows us to confirm, through a statistical test, whether the structure recovered by the mapper graph is indeed present in the point cloud. We use persistent homology, a well-known construction from topological data analysis (TDA), to represent the structure from the point cloud, and a recently introduced hypothesis testing framework [1] that provides a way to evaluate the significance of such a structure. We demonstrate the approach on two examples: a Y-shaped point cloud and a sample of a 3D mesh resembling an ant. These low-dimensional examples allow us to visually inspect the results, laying the groundwork for extensive experiments with higher-dimensional point cloud data used in real-world applications.

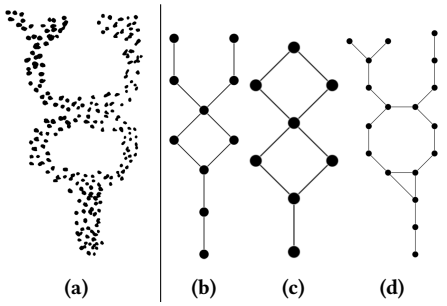


Figure 1: A point cloud (a) and three graphs (b, c, d) summarizing its topological structure, constructed by the mapper algorithm for different choices of its parameters.

Representing the topological structure of the point cloud with a simpler object, such as a graph, and having a statistical method for testing the significance of such a structure is a very relevant task. A simpler representation allows us to visualize [3] and interpret high-dimensional representations that are everywhere in modern data science and machine learning. It might even allow us to find singularities that often carry relevant information. The mapper algorithm [6] is a commonly used tool in TDA. Although it is simple, the result is sensitive to the choice of its parameters [2]. Nevertheless, it provides only one possible low-dimensional view of the input data, and to our knowledge there is no method that would confirm the significance of the represented structure. There is another method, called persistent homology, which, while not directly applicable to visualization, deals with a particular structure of "holes" in space and now has a framework [1] that allows us to statistically test the significance of such a structure.

2 BACKGROUND

A point cloud P is a set of points embedded in $\mathbb{R}^d$ which can be viewed as a sample of a topological space $\mathbb{X}$. Since discrete points from P have no interesting topological structure, we consider the space $P^r = \bigcup_{p \in P} B(p, r)$ for some radius r. If P is a sufficiently dense sample of $\mathbb{X}$, then P^r has some of the same properties as $\mathbb{X}$ for a suitable r. To compute the properties of interest, we represent P^r with a simplicial complex K which, if properly constructed, has homology groups isomorphic to those of P^r. We

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

are interested in finding the branching and cyclic structure in the point cloud, both of which can be detected using (persistent) homology.

2.1 Simplicial complexes

A *(geometric) simplicial complex* K can be thought of as a "high-dimensional graph" whose vertices are points from the point cloud and connectivity is determined by the geometric configuration of the points. In addition to vertices and edges, we include triangles, tetrahedra and higher dimensional simplices. Formally, K consists of finite nonempty subsets of P and is closed under inclusion (i.e., $A \in K$ and $B \subset A$ implies $B \in K$). We refer to elements in K of size $k+1$ as k-simplices, which correspond to k-cliques when we think about K as a hyper-graph.

The Čech and Vietoris-Rips complexes are the two most common constructions, both parameterized by a scale parameter (radius) $r > 0$. We use the Vietoris-Rips construction, where we include a subset of (k + 1) points from P as a k-simplex if all points are at most r apart.

We can construct a sequence of complexes $K_{r_1}, K_{r_2}, \ldots$ by increasing the radius r. Such a construction is "increasing" in the sense that for $r_1 < r_2$, it holds that $K_{r_1} \subseteq K_{r_2}$. Such sequences are also known as *filtrations* and are used in persistent homology.

2.2 Persistent relative homology

Homology. *Homology* is a classical construction in algebraic topology that deals with topological properties of a space. More precisely, it provides a mathematical language for the holes in a topological space. Homology groups denoted by $H_k(\mathbb{X})$, where k is a dimension, capture the holes indirectly by focusing on what surrounds them. For example, the basis of $H_0(\mathbb{X})$ corresponds to the connected components and the basis of $H_1(\mathbb{X})$ to the closed loops surrounding the holes. The rank of the k-th homology group, also known as *Betti number*, counts the number of k-dimensional "holes".

We can construct homology groups for a given simplicial complex K. The important concepts in the construction are: (i) the chain groups C_k, where the k-th chain group consists of all formal linear combinations of k-dimensional simplices $\sum_i a_i \sigma_i$, where σ_i are k-simplices from K and a_i are coefficients, usually from $\mathbb{Z}_2$, (ii) the boundary operator ∂_k, which is a map describing how (k - 1)-simplices are attached to k-simplices, (iii) the groups Z_k of k-cycles, which are k-chains in the kernel of ∂_k, and (iv) the groups B_k of k-boundaries, which are elements in the image of ∂_{k+1}. The boundary operator ∂_k has the property that $\partial_k \circ \partial_{k+1} = 0$, i.e., it maps the boundary of the boundary to zero. Therefore, $B_k \subseteq Z_k$.

Intuitively, a k-cycle can be thought of as a generalized version of a cycle in a graph - it is a sequence of k-dimensional simplices wrapped around something. If this sequence is actually a boundary of a (k+1)-dimensional chain, then its interior is full (trivial cycle). Otherwise, it surrounds a hole. The k-th homology $H_k = ker\ \partial_k / im\ \partial_{k+1} = Z_k / B_k$ takes a "modulo" of k-cycles with k-boundaries, leaving only cycles that are nontrivial.

Relative homology. Given a simplicial complex K and a subcomplex $L \subseteq K$, the relative homology of a pair of topological spaces (simplicial complexes in our case) can be thought of as the (reduced) homology of the quotient space K/L. Intuitively, we want to factor out L, which is expressed by the quotient operation $C_k(K, L) = C_k(K)/C_k(L)$. The group of k-cycles becomes $Z_k(K, L) = Z_k(K)/Z_k(L)$, which we call the group of *relative cycles*. We can think of the reduced homology of a space as if we were representing the entire L with a single point.

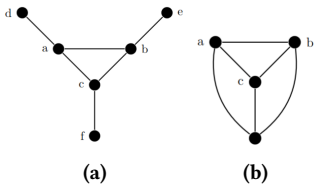


Figure 2: a) A Y-shaped simplicial complex with one cycle. b) The quotient K/L, where subcomplex L contains 0-simplices {d, e, f}. Such identification introduces two new 1-dimensional "holes", captured by the relative homology group $H_1(K, L)$.

The concept of homology and relative homology is best illustrated by an example. Consider a simple simplicial complex consisting of 0-simplices {a, b, c, d, e, f} and 1-simplices {(a, b), (a, c), (a, d), (b, e), (c, f)} as shown in Figure 2a. There is a "hole" of dimension 1 (surrounded by the cycle $a \to b \to c \to a$), which is captured in the homology group H_1. Choosing $L = \{d, e, f\}$ as a subcomplex, the quotient K/L identifies the simplices from L to a single point, as shown in the figure 2b. This results in two new "holes" in dimension 1, which are captured by the relative homology group $H_1(K, L)$, which has rank 3. This "lifting property" of relative homology (introducing new "holes" when identifying simplices) is used in our approach to detect branching points.

Persistent homology. The constriction of the simplicial complex and hence the groups H_k are highly sensitive to the choice of radius r. To overcome this, persistent homology considers the entire range of scales and tracks the evolution of k-cycles as the value of r increases, thus forming a sequence of filtrations. In this process, cycles are created (born) and later filled-in (die). This information is most often represented by *persistence diagrams*, a two dimensional scatter plot, $dgm_k = \{p1, \ldots, p_m\}$, where each point $p_i = (b_i, d_i)$ represents the birth and death times (radius) of the associated persistent cycle.

2.3 Significance testing of persistent cycles

The significance of topological features is often measured by the lifetimes of persistent cycles, i.e., $\delta = (d_i - b_i)$. Although this method is intuitive as it captures the geometric “size” of topological features, [1] uses the statistic $\pi_i = d_i/b_i$. They present a statistical test to determine for each point $p_i \in dgm_k$ whether it is a signal or noise, i.e., a significant structure or the result of noise and randomness in the data. They introduce a special transformation $l(p_i)$ applied to each point from the diagram where the values of $l(p_i)$ follow a certain (LGumbel) distribution if p_i are points corresponding to noisy cycles, while cycles significantly deviating from this distribution are declared as signal. The signal part of dgm_k can be recovered as $dgm_k^s(\alpha) = \{p \in dgm_k : e^{-e^{l(p)}} < \frac{\alpha}{|dgm_k|}\}$ given a p-value α.

Computing persistent homology for an entire filtration is often intractable, as higher values of r lead to a large number of simplices. The common practice is to set a threshold r_{max} and calculate $dgm_k(r_{max})$ using simplices generated up to r_{max}. This often leads to cycles that are "infinite", i.e., born prior to r_{max} but die after r_{max}. The framework also provides an algorithm to

determine the infinite cycles that are already significant, and provides means to select the next r_{max} threshold to inspect infinite cycles that have not yet been determined to be significant.

2.4 The mapper algorithm

Given the topological space $\mathbb{X}$ and a continuous function $f : \mathbb{X} \to \mathbb{R}$, the mapper algorithm [6] constructs a graph $G = (V, E)$ that captures the topological structure of $\mathbb{X}$. It does so by pulling back a cover $\mathcal{U}$ of the space $f(\mathbb{X})$ to a cover on $\mathbb{X}$ through f. We can view the function f and the cover $\mathcal{U}$ as the lens through which the input data $\mathbb{X}$ is examined.

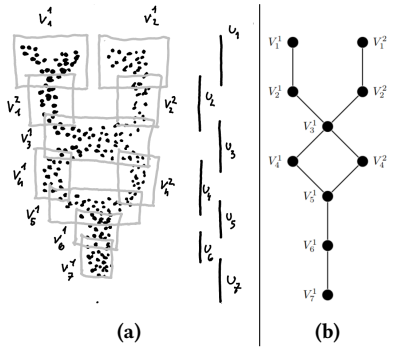


Figure 3: An example of the construction of a mapper graph. (a) A 2-dimensional point cloud P with cover $\{V_i^j\}$, a function $f : \mathbb{R}^2 \to \mathbb{R}$ and cover $\mathcal{U}$ of $f(P)$. (b) The resulting mapper graph.

Given a point cloud P and $f : P \to \mathbb{R}$, we first construct a set of n intervals $\mathcal{U} = \{U_1, \ldots, U_n\}$ covering $f(P)$. The percentage of overlap for two consecutive intervals U_i and U_{i+1} is determined by the parameter p. For each interval $U_i = (a, b)$, let $P_{U_i} = f^{-1}(U_i)$ be a set of points with function values in the range (a, b). The set P_{U_i} for each U_i is further partitioned into $V^1, \ldots, V^{k_i}$ by a clustering algorithm (in our case DBSCAN [5] with parameter ϵ, which sets the maximum distance between two samples so that one is considered to be in the neighborhood of the other) to obtain a cover of $P = \bigcup_{i=1,\ldots,n}\{V_i^1, \ldots, V_i^{k_i}\}$. Each $V_i^j \subset P$ becomes some vertex v in the mapper graph with $\phi(v) = V_i^j$ mapping v to a subset of points. Two vertices are connected by an edge if their point sets intersect (see Figure 3).

The resulting graph $G = (V, E)$ provides a combinatorial description of the data and the mapping $\phi : V \to \mathcal{P}(P)$ maps each node $v \in V$ to a subset of points from P.

3 METHODOLOGY

The input to our approach is a set of points P embedded in $\mathbb{R}^d$ and a graph $G = (V, E)$ together with a mapping $\phi : V \to \mathcal{P}(P)$ that maps each vertex to a subset of points. Note that the method used to construct the graph is not limited to the mapper algorithm.

The graph is assumed to capture the topological structure of the point cloud, i.e., branching points (vertices with a degree of at least 3) and cycles in the graph should reflect the branching and cyclic structure of the point cloud. Our approach tests whether the captured structure is significant when viewed through homology, operating directly on a subset of points from the point cloud.

3.1 Testing the cycles

A *simple cycle* is a finite sequence of vertices $v_1 \to v_2 \to \ldots \to v_n$, where v_i and v_{i+1} are connected by an edge such that no vertex, except the endpoint, repeats ($v_i = v_j$ if and only if $i, j \in \{1, n\}$). Let $v_1, \ldots, v_n$ be such a cycle from G. We compute the persistence diagram of the subset $P' = \bigcup_{i=1,\ldots,n} \phi(v_i)$ and use the test [1] to confirm that it contains at least one significant cycle ("hole") of dimension 1.

3.2 Testing the branching structure

Let $N(v)$ be a set of vertices connected to v (1-hop neighborhood) and let v be a branching point in G (as in Figure 4). Let $N'(v) = \{u : u \in N(v), deg(u) \geq 2\}$ be a set of vertices from $N(v)$ that have at least one additional neighbor. Together with v, $N'(v)$ forms a set of internal points $I_v = \bigcup_{u \in \{v\} \cup N'(v)} \phi(u)$ (shown in Figure 4 as black vertices inside the outer black line).

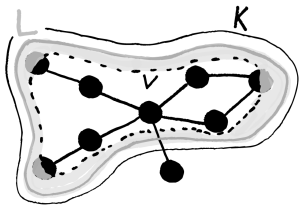


Figure 4: Construction of K and L for a branching point v. Vertices forming K are inside the outer black line. Vertices forming L are bicolored, indicating that some of their points are inside due to overlap between the vertices' point sets.

Let $K_v = \bigcup_{u \in N'(v)} N(u)$ be a set of vertices whose points are used to form a complex K (vertices inside the outer black line in Figure 4), i.e. K is formed from the points $\bigcup_{u \in K_v} \phi(u)$. Now let L be a subcomplex of K containing simplices which do not contain any of the points from I_v. Thus L contains points of vertices exactly two edges away from v (bicolored vertices in Figure 4). We use K and L to compute relative persistent homology, identifying simplices of L to a single point and introducing relative cycles ("holes") when $K \backslash L$ has a branching structure. For a branching point v, the relative persistence diagram should contain at least $deg(v) - 1$ significant relative cycles.

4 EXPERIMENTS

We perform experiments illustrating our approach on two point clouds. The graphs are constructed using the mapper algorithm from the Giotto TDA library [7] with the parameters specified for each experiment. To construct the simplicial complex and compute (relative) persistent homology, we use the Dionysus library [1]. We increase the initial radius r using the algorithm from [1] until either no infinite cycles remain or all currently infinite cycles are identified as significant.

We include a figure of the graph for each experiment and mark interesting branching points and cycles. The points corresponding to a cycle are shown in red, the internal points of a branching point are also red, while the boundary points (forming L) are blue.

[1] Available at: https://github.com/mrzv/dionysus.

4.1 Experiment 1: Y-shaped point cloud

The point cloud P consists of 5000 points in $\mathbb{R}^2$ and resembles a Y-shape with a cycle in the centre. The graph (see Figure 5) was created with the following parameters: f is a projection on the x-coordinate, $n = 30$, $p = 0.5$ and $\epsilon = 3$.

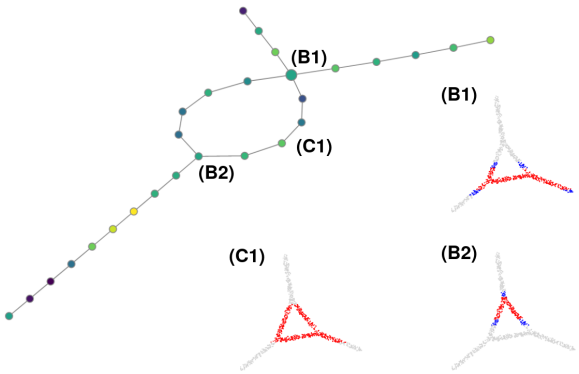


Figure 5: Mapper graph with two branching points (B1 and B2) and one simple cycle (C1) together with their corresponding subsets of points.

The graph contains one simple cycle, which is also significant because the subset of its points contains a homologically significant cycle. The graph also contains two branching points, B1 and B2 with degrees 4 and 3.

The persistence diagram for B1 has three (significant) infinite cycles, indicating a branching structure of degree 4, while the diagram for B2 has two (significant) infinite cycles, indicating a branching structure of degree 3. In this example, it was confirmed that both the cyclic and the branching structure of the graph are reflected in the point cloud.

4.2 Experiment 2: 3D ant surface

The point cloud P consists of 6370 points in $\mathbb{R}^3$ corresponding to the vertices of a 3D mesh in the form of an ant obtained from [4]. The graph (see Figure 6) was created with the following parameters: f is the distance to the tip of the ant's abdomen, $n = 50$, $p = 0.5$, and $\epsilon = 0.025$.

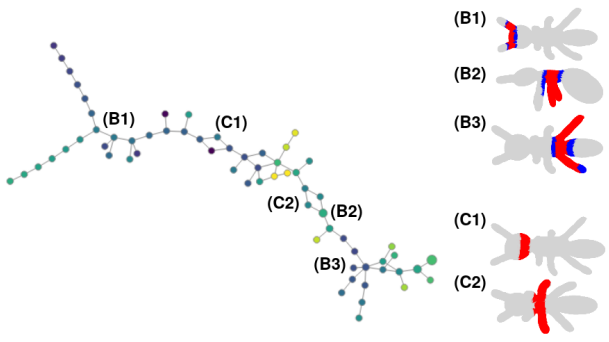


Figure 6: Mapper graph with three highlighted branching points (B1, B2 and B3) and two simple cycle (C1, C2) together with their corresponding subsets of points.

We highlight three interesting branching points. Vertex B1 is a branching point of degree 3, which corresponds to the branching on the ant's head into its two antennas and is significant. Vertex B2 is a branching point of degree 3 and one of the vertices from the cycle C2. Looking at the point cloud, no branching structure is detected because the points of the two legs are contained in the vertex B2 itself and there are no boundary points on the legs, so they appear as a single connected blob. Our approach does not detect a branching structure, even though there is, as some other strategy of selecting the boundary points would need to be used. Vertex B3 has degree 6, but only 5 neighbors are used as one does not have any additional neighbor except B3. Since one of the legs has no boundary points, only 2 cycles appear, causing B3 to be recognized as a branching point with degree 3.

We also highlight 2 simple cycles. Cycle C1 wraps around the ant's hollow head and is recognized as significant. Cycle C2 wraps around the ant's two middle legs and part of its body. No significant cycles were found - ant's legs are not close enough together to form a large cycle and cycle formed by the hollow legs is too small to be detected. So there is not enough support to confirm the structure found by mapper.

5 CONCLUSIONS AND FUTURE WORK

We have demonstrated, how persistent (relative) homology can be used in conjunction with a statistical test to confirm the significance of the topological structure of a point cloud summarized with a graph. In the future, we will conduct extensive experiments on more complex, high-dimensional point clouds with known and unknown structure. Ideally, we could use our approach to prune the mapper graphs or guide the selection of values for its parameters. Our approach to identifying branching structures needs further work, as the current strategy of using a (modified) 2-hop neighborhood as a boundary sometimes fails. In addition, we may need a more sensitive version of the statistical test from [1] which is currently stated to hold in general but might be possible to adapt for a particular type of data.

ACKNOWLEDGEMENTS

This work was supported by the Slovenian Research Agency under the project J2-1736 Causalify and co-financed by the Republic of Slovenia and the European Union's HE program under enRichMyData EU project grant agreement number 101070284.

REFERENCES

[1] Omer Bobrowski and Primoz Skraba. 2023. A universal null-distribution for topological data analysis. *Scientific Reports*, 13, 1, (July 2023), 12274. DOI: 10.1038/s41598-023-37842-2.

[2] Mathieu Carrière, Bertrand Michel, and Steve Oudot. 2018. Statistical analysis and parameter selection for mapper. *Journal of Machine Learning Research*, 19, 12, 1–39. http://jmlr.org/papers/v19/17-291.html.

[3] Nithin Chalapathi, Youjia Zhou, and Bei Wang. 2021. Adaptive covers for mapper graphs using information criteria. In *2021 IEEE International Conference on Big Data (Big Data)*, 3789–3800. DOI: 10.1109/BigData52589.2021.9671324.

[4] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. 2009. A benchmark for 3d mesh segmentation. *ACM Trans. Graph.*, 28, 3, Article 73, (July 2009), 12 pages. DOI: 10.1145/1531326.1531379.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In (KDD'96). AAAI Press, Portland, Oregon, 226–231.

[6] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Eurographics Symposium on Point-Based Graphics*. M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors. The Eurographics Association. ISBN: 978-3-905673-51-7. DOI: 10.2312/SPBG/SPBG07/091-100.

[7] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Anibal Medina-Mardones, Alberto Dassatti, and Kathryn Hess. 2020. Giotto-tda: a topological data analysis toolkit for machine learning and data exploration. (2020). arXiv: 2004.02551 [cs.LG].

Highlighting Embeddings' Features Relevance Attribution on Activation Maps

Jože M. Rožanec
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
joze.rozanec@ijs.si

Erik Koehorst
Philips Consumer Lifestyle BV
Drachten, The Netherlands
Erik.Koehorst@philips.com

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

The increasing adoption of artificial intelligence requires a better understanding of the underlying factors affecting a particular forecast to enable responsible decision-making and provide a ground for enhancing the machine learning model. The advent of deep learning has enabled super-human classification performance and eliminated the need for tedious manual feature engineering. Furthermore, pre-trained models have democratized access to deep learning and are frequently used for feature extraction. Nevertheless, while much research is invested into creating explanations for deep learning models, less attention was devoted to how to explain the classification outcomes of a model leveraging embeddings from a pre-trained model. This research focuses on image classification and proposes a simple method to visualize which parts of the image were considered by the subset of the most relevant features for a particular forecast. Furthermore, multiple variants are provided to contrast relevant features from a machine learning classifier and selected features during a feature selection process. The research was performed on a real-world dataset provided by domain experts from *Philips Consumer Lifestyle BV*.

KEYWORDS

explainable artificial intelligence, feature importance, activation map, GradCAM, image classification, smart manufacturing, defect detection

1 INTRODUCTION

The increasing adoption of artificial intelligence has posed new challenges, including enforcing measures to protect the human person from risks inherent to artificial intelligence systems. One step in this direction is the European AI Act [12], which considers that different artificial intelligence systems must conform to a different set of requirements according to their risk level, linked to the particular domain and potential impact on health, safety, or fundamental rights [15]. In this context, explainable artificial intelligence, a sub-field of machine learning, has gained renewed attention with the advent of modern deep learning [22], given that it researches how more transparency can be brought to opaque machine learning models. While transparency in the regulatory context is sought to enable responsible decision-making, it provides valuable insights to enhance the workings of machine learning models, too.

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

The field of explainable artificial intelligence can be traced back to the 1970s [18]. A key question posed by the researchers is what makes a good explanation. Arrieta et al. [2] consider that a good explanation must take into account at least three elements: (a) the reasons for a given model output (e.g., features and their value ranges), (b) the context (e.g., context on which inference is performed), and (c) how are (a) and (b) conveyed to the target audience (e.g., what information can be disclosed and the vocabulary used, among others). When considering images, maps frequently present explanations that contrast particular model information on top of the original input image (e.g., saliency maps, activation maps, heat maps, or anomaly maps [13, 24]). Other approaches can be extracting and highlighting super-pixels relevant to a specific class [16] or the occlusion of background parts irrelevant to the model. Such outputs convey (a) the reasons for a given model output by highlighting the images, (b) the context on which inference is performed (by overlaying the information on top of the image used for inference), and (c) using an agreed approach to convey to the user what is considered more relevant and what is not.

Multiple approaches have been developed to explain the inner workings of image classifiers. LIME (Local Interpretable Model-Agnostic Explanations) [16] approached this challenge by retrieving predicted labels for a particular class and showing the segmented superpixels that match each class. GradCAM[19] has taken another approach and created activation maps considering the weight of the activations at particular deep learning model layers by the average gradient. Many approaches were developed afterward, following the same rationale. For example, GradCAM++[3], XGradCAM[9], or HiResCAM[6] work like GradCAM but consider second-order gradients, scale the gradients by the normalized activations, or element-wise multiply the activations with the gradients respectively. Other possible approaches are leveraging insights resulting from image perturbation [8] or methods that acquire and display samples similar or counterfactual to the predicted instance [4, 17].

The development of information and communications technologies fostered the emergence of the Industry 4.0 paradigm as a technology framework to integrate and extend manufacturing processes [23]. In this context, the increasing adoption of artificial intelligence enables greater automation of manufacturing processes such as defect inspection [7] and urges the adoption of explainable artificial intelligence to develop users' trust in the models and foster responsible decision-making based on the insights obtained regarding the underlying machine learning model [1].

From the literature mentioned above and several surveys on this topic [5, 13, 14, 17, 20, 21], it was found that the authors did not contemplate how explanations can be provided in scenarios where feature embeddings are extracted with a deep learning model and then used to train a separate machine learning model.

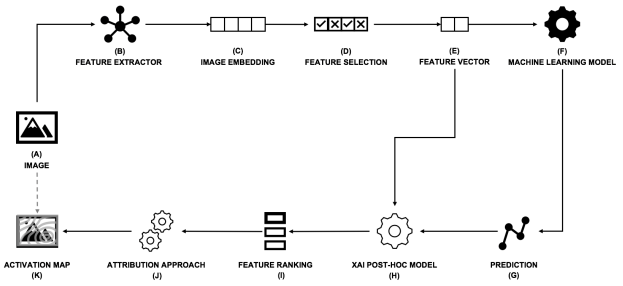


Figure 1: To classify an image, a feature extractor is used to create an embedding, from which certain values are extracted to create a feature vector. The machine learning model issues a prediction, which, along with the feature vector, is used to create a feature ranking. The attribution approach considers the highest-ranking features to generate an activation map.

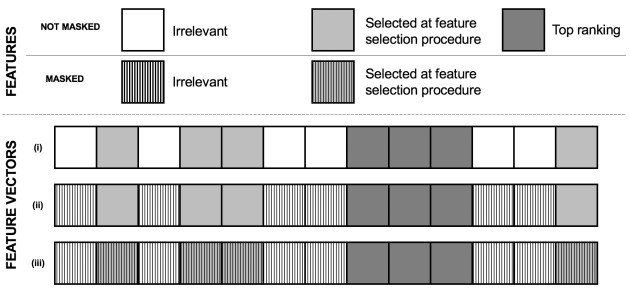


Figure 2: Given an image embedding (i), we can mask it to display (ii) features selected at the feature selection procedure (including the top ranking classifier's features, or (iii) can mask it to display only the top ranking classifier's features.

Figure 3: Sample images from the dataset provided by *Philips Consumer Lifestyle BV*. Three categories are distinguished: images corresponding to non-defective items (good) and images corresponding to two defect types (double-printed and with interrupted prints).

The present research addresses this void by proposing an unsupervised approach to generate activation maps based on the feature ranking obtained for a particular forecast. The research is performed on a real-world dataset provided by *Philips Consumer Lifestyle BV* and related to defect inspection.

This paper is organized as follows. First, section 2 describes the explainability approach developed and tested in this research. Section 3 describes the experiments performed to assess different value imputation strategies, and Section 4 informs and discusses the results obtained. Finally, Section 5 concludes and describes future work.

2 HIGHLIGHTING EMBEDDINGS' FEATURES RELEVANCE ATTRIBUTION ON ACTIVATION MAPS

The increasing amount of pre-trained deep learning models make them the default choice for feature extraction when working with machine learning models for images. Nevertheless, the disconnect between the machine learning model built on top and the deep learning model used to extract the image embedding makes it challenging to provide good explanations to the user. This research proposes an approach to bridge the gap (see Fig. 1). In particular, we leverage the fact that similar images or fragments of images result in embeddings or parts of embeddings that are close to each other. This property can be exploited when building activation maps, computing the similarity between a reference image (e.g., the image of a horse) and the image under consideration to find where such class can be found in the image under consideration (e.g., given the image of a farm, highlight where the horses are located). Nevertheless, if instead of using some reference image, the image that is an input to the machine learning model is leveraged as a reference, (i) no noise is introduced due to the dissimilarity of the images, and (ii) no beforehand knowledge regarding the classes of interest is required. Therefore, a key issue must be resolved: how do both embeddings differ to ensure that such difference is exploited to build an activation map?

Two options are envisioned in this research (see Fig. 2): given (i) the image embedding, two variations can be considered for value imputation: (ii) mask all the values in the embedding except for the ones corresponding to top-ranking features, (iii) mask all the values in the embedding except for the ones corresponding to selected features and top-ranking features, using different values for each of them. By doing so, the highest similarity in the image will be found in regions related to top-ranking features or selected features. Considering selected and top-ranking features provides additional insights into what information was provided to the model and what information was considered the most important by the model. These two approaches are explored in Section 3.

3 EXPERIMENTS

We experimented with a real-world dataset of logos printed on shavers provided by *Philips Consumer Lifestyle BV*. The dataset consisted of 3518 images considered within three categories (see Fig. 3): non-defective images and images with two kinds of defects (double-printed logos and interrupted prints). To extract features from the images, the ResNet-18 model [10] was used, extracting the features before the fully connected layer. Mutual information was used to evaluate the most relevant features and select the *top K*, with $K = \sqrt{N}$, where N is the number of data instances in the train set, as suggested in [11]. The dataset was divided into train (75%) and test (25%), and a random forest classifier was trained on it, achieving an AUC ROC (one-vs-rest) score of 0.9022.

Three images from the test set were considered for the experiments: good, double-printed, and with an interrupted print. The images were randomly picked among the available ones for that particular class. To assess the features' relevance of a particular forecast, LIME[16] was used, considering the top 1, 3, 5, 7, and 13 ranked features.

Figure 4: GradCAM activation maps for ResNet-18 layers 1-4 and four layers combined.

The GradCAM images were generated for ResNet-18 layers 1-4 and another image considering the four layers. To understand where the underlying model focused, we created GradCAM activation maps contrasting the image against itself (see Fig. 4). The cosine similarity between the imputed vector and the image embedding was computed across test samples (880 samples: 679 good, 58 double-printed, and 143 related to interrupted printing). The mean similarity and standard deviation were used to assess whether the imputation strategy increased the similarity or contrast between the imputed vector and the image embedding.

The GradCAM images were generated by computing the cosine similarity between the image embedding and the feature vector generated considering three strategies described in Table 1. A sample of the resulting activation maps were visually assessed and are reported in Section 4.

The experiments were designed to understand which imputation strategy works the best. A detailed analysis regarding how top-ranked features affect the activation maps was omitted due to the brevity of the paper.

Strategy	Top-ranked feature	Selected on Feature Selection	Irrelevant
TOZ	True value	One	Zero
TZZ	True value	Zero	Zero
TRR	True value	Random	Random

Table 1: Value imputation strategies considering the image embedding, the features selected during the feature selection process, and the classifier's top-ranked features.

4 RESULTS

Imputation strategy	Image class	Layers			
		1	2	3	4
TOZ	Good	0.27±0.01	0.27±0.01	0.27±0.01	0.27±0.01
	Double-printed	0.31±0.02	0.31±0.02	0.31±0.02	0.31±0.02
	Interrupted print	0.27±0.01	0.27±0.01	0.27±0.01	0.27±0.01
TZZ	Good	0.21±0.04	0.21±0.04	0.21±0.04	0.21±0.04
	Double-printed	0.24±0.03	0.24±0.03	0.24±0.03	0.24±0.03
	Interrupted print	0.22±0.04	0.22±0.04	0.22±0.04	0.22±0.04
TRR	Good	0.46±0.02	0.46±0.02	0.46±0.02	0.46±0.02
	Double-printed	0.48±0.03	0.48±0.03	0.48±0.03	0.48±0.03
	Interrupted print	0.46±0.02	0.46±0.02	0.46±0.02	0.46±0.02

Table 2: Value imputation strategies considering the image embedding, the features selected during the feature selection process, and the classifier's top-ranked features.

As described in Table 1, three imputation strategies were considered. The cosine similarity computed between the vector created with the imputation strategy and the embedding (considering the top 13 features) is reported in Table 2. A higher similarity between the imputed vector and the image embedding means that a wider area of the activation map will be highlighted, blurring relevant information where the top features point to in the image. The less informative imputation strategy was TRR, which consistently showed high cosine similarity across layers for all defect types. On the other hand, TZZ achieved the best results regardless of the defect and layer considered. Imputing selected features with one had a detrimental effect, given it increased the similarity between the imputed vector and the embedding. Nevertheless, the similarity was usually between 0.10 and 0.20 points below that reported with the TRR imputation strategy.

For visual assessment, activation maps for different imputation strategies obtained for the top 13 features are displayed in Fig. 5. When comparing TZZ and TRR strategies, we found that for layer one, TZZ for the double-printed image focused on the top contour of characters, and for the interrupted print highlighted regions of relevance. In contrast, TRR did not highlight any region for the double-printed image and highlighted fewer regions for the interrupted print when compared to TZZ. For layer two, TZZ for the image of the non-defective product displayed some artifacts but included areas covering characters' contours, too. Furthermore, for the double-printed and interrupted print images, it covered relevant regions. TRR, on the other hand, highlighted different regions, which, for the good and double-printed images, were mostly irrelevant. For layer three, TZZ highlighted mostly irrelevant areas for the image of the non-defective product, except for the character "S". For the double-printed image, the beginning and end of the words are highlighted, while for the interrupted prints, the highlighted areas covered places where defects were observed. TRR, on the other hand, for the good image, covered two-thirds of the image, and for the double-printed, it highlighted most of the areas highlighted with the PZZ strategy. Nevertheless, for the interrupted print, most focus was placed on the lower part of the "P" char, while also two artifacts were encountered. Finally, for the fourth layer, TZZ has mostly focused on the upper word (Philips), while TRR's focus was mostly on the lower part of the image, still covering some relevant areas.

When comparing the TZZ and TOZ approaches, we found that for layer one, TOZ results in less strongly highlighted regions: most of the highlighted regions present in TZZ vanished, and just in the good image, a few spots appeared that were not present at the TZZ activation map. The original regions are highlighted for layer two, but new regions were included, mostly covering areas of interest. The highlighted areas for a double-printed image related to TZZ and TOZ activation maps were consistent for layer three. Nevertheless, TOZ highlighted different regions for the good and interrupted print images. The regions highlighted for the interrupted print image were irrelevant to defect detection. When considering the last layer, the highlighted areas were mostly the same for TZZ and TOZ. Nevertheless, an additional region was introduced in the good and interrupted print images, covering the lower text.

From the visual assessment described above, we conclude that activation maps obtained with the TZZ imputation method lead to the best explanations.

5 CONCLUSIONS

This work has researched how information regarding feature importance when using image embeddings can be used and propagated back to generate activation maps and highlight regions of the image considered relevant to a particular forecast. The proposed approach was evaluated on images of a real-world industrial use case. The similarity metrics and visual evaluation show that the best value imputation strategy is TZZ, which considers assigning the actual embedding value to relevant features

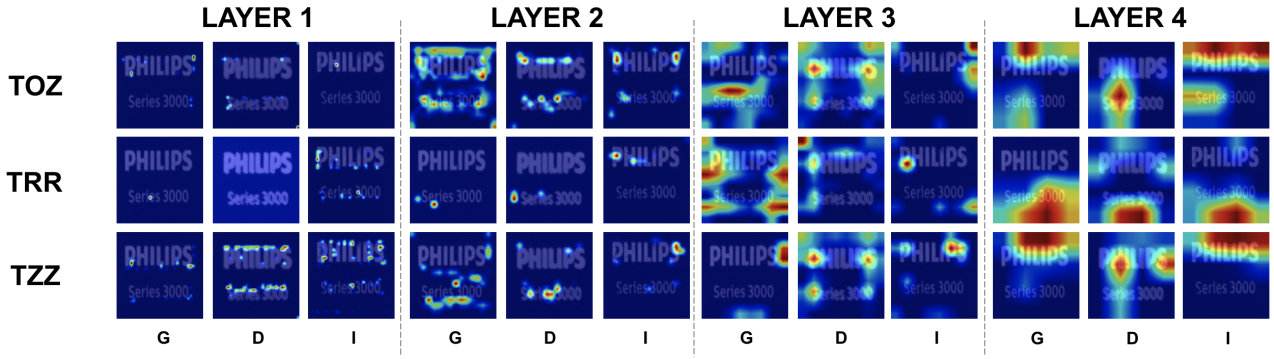


Figure 5: GradCAM activation maps for ResNet-18 layers 1-4 considering only the top 13 features for this particular forecast and three imputation strategies (TOZ, TZZ, and TRR) for three image types (good (G), double-printed (D), and interrupted prints (I)).

and masking the rest of the embedding with zeroes. Nevertheless, it must be emphasized that a broader set of experiments must be considered to generalize these conclusions. While this research only considered local explanations, the feature relevance could be considered at a global level, and the same approach was leveraged to visualize their influence on a particular image. Future work will focus on a more comprehensive evaluation of the proposed methodology to understand how it performs, how the number of selected features influences the activation maps and possible shortcomings.

ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the European Union's Horizon 2020 program project STAR under grant agreement number H2020-956573.

REFERENCES

[1] Imran Ahmed, Gwanggil Jeon, and Francesco Piccialli. 2022. From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. *IEEE Transactions on Industrial Informatics* 18, 8 (2022), 5031–5042.

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion* 58 (2020), 82–115.

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[4] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems* 32 (2019).

[5] Arun Das and Paul Rad. 2020. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371* (2020).

[6] Rachel Lea Draelos and Lawrence Carin. 2020. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. *arXiv preprint arXiv:2011.08891* (2020).

[7] Gautam Dutta, Ravinder Kumar, Rahul Sindhwani, and Rajesh Kr Singh. 2021. Digitalization priorities of quality control processes for SMEs: A conceptual study in perspective of Industry 4.0 adoption. *Journal of Intelligent Manufacturing* 32, 6 (2021), 1679–1698.

[8] Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*. 3429–3437.

[9] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. 2020. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312* (2020).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Jianping Hua, Zixiang Xiong, James Lowey, Edward Suh, and Edward R Dougherty. 2005. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics* 21, 8 (2005), 1509–1515.

[12] Tambiama André Madiega. 2021. Artificial intelligence act. *European Parliament: European Parliamentary Research Service* (2021).

[13] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. 2022. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review* (2022), 1–66.

[14] Sajid Nazir, Diane M Dickson, and Muhammad Usman Akram. 2023. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine* (2023), 106668.

[15] Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, et al. 2023. The role of explainable AI in the context of the AI Act. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 1139–1150.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[17] Gesina Schwalbe and Bettina Finzel. 2023. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery* (2023), 1–59.

[18] A Carlisle Scott, William J Clancey, Randall Davis, and Edward H Shortliffe. 1977. *Explanation capabilities of production-based consultation systems.* Technical Report. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

[19] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[20] Bas HM Van der Velden, Hugo J Kuijf, Kenneth GA Gilhuijs, and Max A Viergever. 2022. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis* 79 (2022), 102470.

[21] Giulia Vilone and Luca Longo. 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion* 76 (2021), 89–106.

[22] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*. Springer, 563–574.

[23] Li Da Xu, Eric L Xu, and Ling Li. 2018. Industry 4.0: state of the art and future trends. *International journal of production research* 56, 8 (2018), 2941–2962.

[24] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8330–8339.

An approach to creating a time-series dataset for news propagation: Ukraine-war case study

Abdul Sittar
abdul.sittar@ijs.si
Jožef Stefan Institute and Jožef Stefan Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute and Jožef Stefan Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

ABSTRACT

An efficient technique to comprehend news spreading can be achieved through the automation of machine learning algorithms. These algorithms perform the prediction and forecasting of news dissemination across geographical barriers. Despite the fact that news regarding any events is generally recorded as a time-series due to its time stamps, it cannot be seen whether or not the news time-series is propagating across geographical barriers. In this article, we explore an approach for generating time-series datasets for news dissemination that relies on Chat-GPT and sentence-transformers. The lack of comprehensive, publicly accessible event-centric news databases for use in time-series forecasting and prediction is another limitation. To get over this bottleneck, we collected a news dataset consisting of 1 year and 3 months related to the Ukraine war using Event Registry. We also conduct a statistical analysis of different time-series (propagating, unsure, and not-propagating) of different lengths (2, 3, 4, 5, and 10) to document the prevalence of geographical barriers. The dataset is publicly available on Zenodo.

KEYWORDS

news propagation, time-series dataset, geographical barriers, Ukraine-war

1 INTRODUCTION

The process of information traveling from a sender to a set of receivers via a carrier is commonly referred to as propagation [3]. News propagate over time by different publishers about an event. It implicitly raises a few thoughts in our mind, such as: 1) There will be some news articles propagating similar information over time; 2) some news articles will be of a unique category that eventually will not be propagating or propagating across geographical barriers by a few publishers.
News streaming is classified into events where a relevant set of news is clustered and represented as an event [8, 9]. And there is a starting and ending time for an event, which is calculated by the publication time of the first and last news article. Hence, an event consists of a set of news articles, and these news articles follow a certain pattern based on hidden properties including cultural, economical, political, linguistic, and geographical [17].

Information Society 2023, 10 October 2023, Ljubljana, Slovenia

Moreover, news spreading comes across many barriers due to different reasons, including cultural, economic, political, linguistic, or geographical, and these reasons depend upon the type of news, such as sports, health, science, etc. [18]. For instance, it is more likely that the news spreading relating to the FIFA World Cup crosses cultural barriers since it involves multiple cultures. Similarly, news spreading relating to the Sri-Lankan economic crisis and the Ukraine-war probably comes across economic and geographical barriers since these events involve multiple stances from the international community; Eid celebrations and Christmas are likely to come across religious barriers; US elections are likely to come across political barriers [17].
The identification of news spreading patterns while crossing barriers can be useful in the context of numerous real-world applications, such as trend detection and content recommendations for readers and subscribers. To perform the classification of news published across barriers (geographical, cultural, economic, etc.) and, in that attempt, to recommend and identify trends of news spreading belonging to different categories, some methodological considerations are necessary.
In this paper, we introduce an approach to creating a time-series dataset for news propagation. While previous work has focused on creating events from collections of news articles [9, 16], we focus on creating propagation time-series. We take the Ukraine-war as an example to be researched in the propagation analysis across geographical barriers.
Following are the main scientific contributions of this paper:

(1) We present an approach to creating a time-series dataset for news propagation.
(2) A dataset for forecasting and predicting news propagation, that has been labeled with the assistance of Chat-GPT and sentence transformers.

The remainder of the paper is structured as follows. Section 2 describes the related work on barriers to news spreading, time-series datasets for news propagation, and topic modeling. Section 3 presents the proposed approach. We discuss the dataset construction and annotation guidelines in Section 4. The evaluation details and statistical analysis is explained in Section 5, while Section 6 concludes the paper and outlines areas of future work.

2 RELATED WORK

In this section, we review the related literature about geographical barriers to news spreading, time-series datasets for news propagation, and topic modeling.

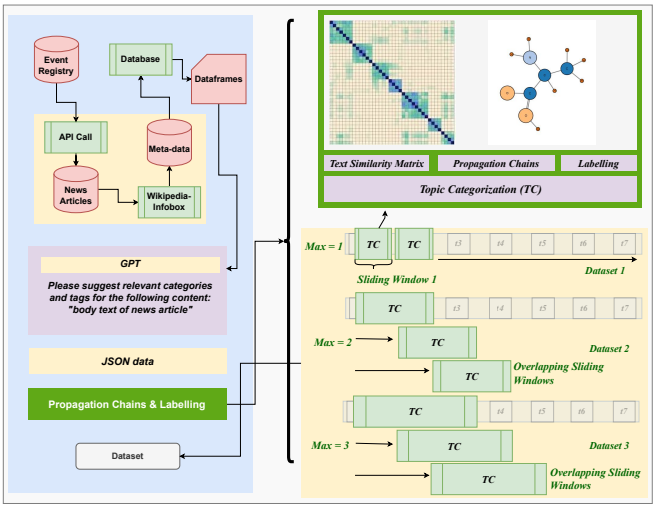


Figure 1: An overview of the proposed approach. To create the propagation time-series, it calculates the semantic similarity across news utilizing sentence transformers, and to evaluate the labeling process of the news, it utilizes a summary of the news articles generated by Chat-GPT.

2.1 Geographical barrier

Sittar reported that the geographical size of a news publisher's country is directly proportional to the number of publishers and articles reporting on the same information [17]. It is also reported that, based on some factors, the media targets specific foreign and regional events. For example, the spreading of news related to specific events may tilt toward developed countries such as the United Kingdom, the U.S.A., or Russia. Also, in the past, geographical representation of entities and events has been extensively utilized to detect local, global, and critical events [10, 20, 19, 2]. It has been said that countries with close distance share culture and language up to a certain extent, which can further reveal interesting facts about shared tendencies in information spreading [12, 11]. Given the difficulty of gathering longitudinal data, relatively little news flow research has systematically examined whether and to what extent foreign nation visibility and the factors that influence it have changed over time. Specifically, scholarship has typically only addressed why some countries get more news coverage than others at a specific point in time, not how and why the focus shifts over time from one country to another [5]. In this context, we propose an approach to collecting data to analyze the news spreading across geographical barriers.

2.2 Time-series datasets

News propagation can be represented in the form of a time-series [17]. The properties of cascading time-series can tell us the relationship between the time and size of cascading. It further answers which events last over a longer period with large communities across different languages. A time-series dataset can be used to understand evolving discussions over time. Different studies have utilized time-series datasets, such as [1] investigates how different discussions evolved over time and the spatial analysis of tweets related to COVID-19. [14] identifies how the discussions evolved over time in top newspapers belonging to three different continents (Europe, Asia, and North America) and nine different countries (UK, India, Ireland, Canada, the U.S.A., Japan, Indonesia, Turkey, and Pakistan). It uses spatio-temporal topic modeling and sentiment analysis. Different classification or mining tasks are proposed using time-series datasets. [6] has proposed the task of predicting stock market values such as price or volatility based on the news content or derived text features. Similarly, to forecast the values, a set of final classes is already defined, such as up meaning an increase in price, down meaning a decrease in price, and balanced meaning no change in price. Also, the same technique has been applied to predict price trends (incline, decline, or flat) immediately after press release publications. Also, Good news articles are categorized as inclines if the stock price relevant to the given article has increased with a peak of at least three points from its original value at the publication time [13].

2.3 Topic modeling

Generally, to find out the most important topics inside an event, multiple solutions have been proposed, including pooling based LDA and BERTopic. Unlike simple static topic modeling, pooling-based techniques assume that the data is partitioned on a time basis, e.g., hourly or daily. Pooling-based techniques are mostly applied to social media, where documents or tweets are partitioned based on hashtags and authors. BERTopic leverages transformers and TF-IDF to create dense clusters, allowing for easily interpretable topics while keeping important words in the topic descriptions. Therefore, the result is a list of topics ranked according to their importance.

The topic modeling techniques are performing surprisingly well. The relation of such topics to their hidden characteristics, such as cultural, economical, and political, has been analyzed in many studies because understanding its dynamics can help governments disseminate information effectively [4, 17, 14, 15]. It has changed rapidly in recent years with the emergence of social media, which provides online platforms for people worldwide to share their thoughts, activities, and emotions and build social relationships [7]. Over the years, scholars have studied the relationship between the news prominence of a country and its physical, economic, political, social, and cultural characteristics [11]. Communication scholars have long been interested in identifying the key determinants of what makes foreign countries newsworthy and why some countries are considered more newsworthy than others [5].

3 APPROACH

This research article presents an approach to creating a time-series dataset for news propagation across geographical barriers, as shown in Figure 1. In the first step, we call an API that extracts the news articles from the Event Registry belonging to Ukraine-war. In the second step, we extract meta-data related to news publishers via searching for the news publishers on Google and extracting their Wikipedia links. Using these links, we obtain the necessary information from Wikipedia-Infobox [17]. We use the Bright Data service to crawl and parse Wikipedia-Infoboxes. In the third step, we perform the summarization of news articles. In the last step, we create a propagation time-series and perform labeling of

the time-series. To calculate the semantic similarity, we utilize monolingual sentence transformers. Since the propagation of information can be captured in the form of time-series we create time-series of different lengths, such as 2, 3, 4, 5, and 10. To evaluate the labeling process, we manually compare the summary generated by Chat-GPT (see Section 5).

4 DATASET CONSTRUCTION

We collected the news articles reporting on the Ukraine-war. Since Russia invaded Ukraine on February 24, 2022, in an escalation of the Russo-Ukrainian War, we fetched news articles that were published between January 2022 and March 2023. The dataset consists of 61261 news articles. Each news article consists of a few attributes: title, body text, name of the news publisher, date, and time of publication.

4.1 Semantic similarity

We calculate the cosine similarity between dense vector generated by sentence transformers. Sentence Transformers is a Python framework for state-of-the-art sentence, text, and image embeddings. Cosine similarity varies between zero and one; zero means no similarity, and one means maximum similarity, i.e., a duplicate article.

4.2 Chat-GPT Summarizing

Since manual evaluation of propagation time-series is difficult because of the length of the news articles, we utilized Chat-GPT to get the tags, categories, and summary representing the whole article. Summarizing a text is one of the many tasks ChatGPT is extremely good at. We can give it a piece of content and ask for a summary. By customizing our prompts, we can get ChatGPT to create much more than a plain summary. We have used the OpenAI API with the Python library. We used the following prompt to fetch the summary of the text, categories, and tags: "Please summarize the text and suggest relevant categories and tags for the following content: articleText:". articleText is a variable representing the text of a news article.

4.3 Annotations of time-series

We created three types of time-series recursively and annotated them based on a threshold of semantic similarity, as shown in Algorithm **??**. The threshold to decide the type of propagation time-series has been set by manually analyzing the similarity and summary of news articles. We set three thresholds for all three types of labels (propagating, unsure, and not-propagating). For instance, the time-series with greater or equal to 0.7 similarity were labeled "Propagating", the time-series with greater or equal to 0.5 similarity were labeled "Unsure", and the time-series with less than 0.5 similarity were labeled "Not-propagating". This criteria has been followed for the minimum length of a time-series (2). However, for the length of a time-series greater than 2, we count the number of pairs with each label, and then the time-series is labeled as one with the highest count. If two labels have the same highest count, then we give priority to the "Propagating" label over "Unsure" and "Unsure" over "Not-Propagating". The Algorithm **??** takes five parameters, such as the start and end of the data-frames, a copy of the data-frames, length of the time-series, and an array. The statistics about the propagation time-series are presented in Figure 2.

To annotate the propagation time-series across geographical barriers, we consider the label "Propagating" for a pair of news articles if the pair is published from two different countries; otherwise, we label it "Not-Propagating". We repeat this process for all lengths of news articles. The statistics after applying this guideline are presented in Figure 3.

5 STATISTICAL ANALYSIS AND EVALUATION

The statistics about the propagation time-series without taking geographical barriers into account are presented in bar chart 2. The number of time-series with the label "Propagating" is higher than the "Unsure", and "Not-Propagating" labels when the length of the time-series is 3 or 5, whereas in the other three cases (2, 4, and 10), the number of time-series is equal for all three labels. The statistics of the propagation time-series that are generated after taking the geographical location of the news publisher into account are presented in bar chart 3. The number of propagation time-series with "Propagated" and "Unsure" labels reduced to almost 40% whereas the number of propagation time-series with the "Not-propagated" label increased significantly.

For the evaluation of the dataset, we have checked the summary, including categories and tags of articles for a specific label, manually. We randomly selected 50 time-series of different lengths for all three types of labels. According to the manual evaluation, the propagation time-series with the "Propagating" label followed almost one or two themes of discussion for all the news articles in a chain. For instance, the following topics have appeared in the propagation time series of length 5: 1) "The United States will be sanctioning Russian President Vladimir Putin; 2) "the national team of the Polish FA will not play against Russia; 3) the Polish Football Association will not play its World Cup qualifying match against Russia; 4) "the Polish Football Association has refused to play a World Cup against Russia; 5) "the Polish national team does not intend to play-off match against Russia". On the contrary, propagation time-series with "Not-Propagating" labels discussed always different points of view about the Ukraine-war. For example, the following topics have appeared in the propagation time-series of length 5: 1) "a resolution passed against Russia in the United Nations"; 2) "Canadian president urges to impose sanctions against Russia"; 3) "the UN Security Council has voted on a US-led draft resolution; 4) "President Trump is inviting Russian President Vladimir Putin to come to Washington; and 5) "India abstained from the vote on the draft resolution". However, in the case of propagation time-series with "Unsure" labels, there were three or four sub-topics discussing the Ukraine-war.

Evaluation results show that as the window size increased to capture the information propagation, the noise of overlapping topics also increased. Similarly, this overlapping window presented sub-topics that overlapped at the time of publication.

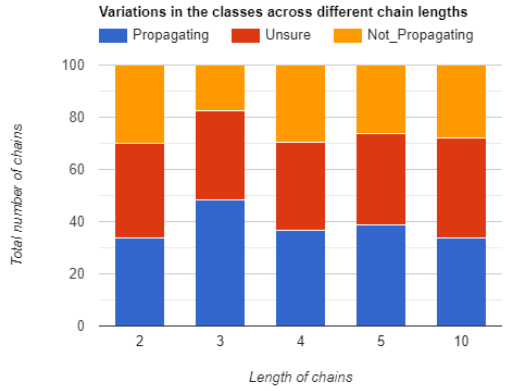


Figure 2: The bar chart shows the statistics about the propagation time-series of different lengths (2, 3, 4, 5, 10) that has been labelled as "Propagating", "Unsure", and "Not-Propagating". The x-axis shows the length of time-series, the y-axis shows the count of the propagation time-series.

.

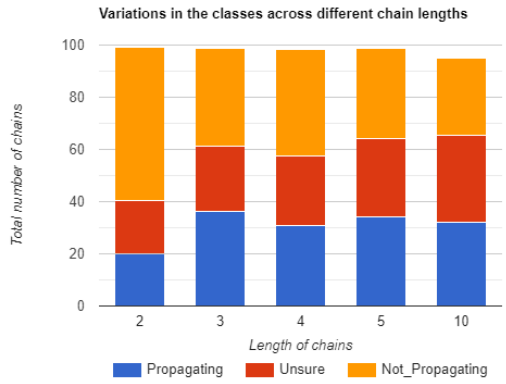


Figure 3: The bar chart shows the statistics about the propagation time-series after applying the condition of the location of a news publisher. Each bar presents three types of propagation time-series that has been labelled as "Propagating", "Unsure", and "Not-Propagating". The x-axis shows the length of time-series, the y-axis shows the count of the propagation time-series.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach to creating a time-series dataset. The goal of this work was to investigate the length of the propagation time-series for news propagation. In the future, we plan to utilize the same approach for different events. Moreover, currently, geographical barriers have been analyzed. In the future, we would like to extend the barriers to political, economic, and cultural barriers and find patterns of news propagation. Also, we would like to perform prediction and forecasting on the labeled time-series dataset. We would like to perform experiments with classical time-series classification methods, deep learning, transformer-based methods, and large language models (LLMs).

ACKNOWLEDGMENTS

The research described in this paper was supported by the Slovenian research agency under the project J2-1736 Causalify and by the EU's Horizon Europe Framework under grant agreement number 101095095.

REFERENCES

[1] Iyad AlAgha. 2021. Topic modeling and sentiment analysis of twitter discussions on covid-19 from spatial and temporal perspectives. *Journal of Information Science Theory and Practice*, 9, 1, 35–53.

[2] Simon Andrews, Helen Gibson, Konstantinos Domdouzis, and Babak Akhgar. 2016. Creating corroborated crisis reports from social media data through formal concept analysis. *Journal of Intelligent Information Systems*, 47, 2, 287–312.

[3] Firdaniza Firdaniza, Budi Nurani Ruchjana, Diah Chaerani, and Jaziar Radianti. 2021. Information diffusion model in twitter: a systematic literature review. *Information*, 13, 1, 13.

[4] Guoyin Jiang, Saipeng Li, and Minglei Li. 2020. Dynamic rumor spreading of public opinion reversal on weibo based on a two-stage spnr model. *Physica A: Statistical Mechanics and its Applications*, 558, 125005.

[5] Timothy M Jones, Peter Van Aelst, and Rens Vliegenthart. 2013. Foreign nation visibility in us news coverage: a longitudinal analysis (1950-2006). *Communication Research*, 40, 3, 417–436.

[6] Abdullah S Karaman and Tayfur Altiok. 2004. An experimental study on forecasting using tes processes. In *Proceedings of the 2004 Winter Simulation Conference, 2004.* Vol. 1. IEEE.

[7] Sanjay Kumar, Muskan Saini, Muskan Goel, and BS Panda. 2021. Modeling information diffusion in online social networks using a modified forest-fire model. *Journal of intelligent information systems*, 56, 2, 355–377.

[8] Haewoon Kwak and Jisun An. 2016. Two tales of the world: comparison of widely used world news datasets gdelt and eventregistry. In *Proceedings of the International AAAI Conference on Web and Social Media* number 1. Vol. 10, 619–622.

[9] Gregor Leban, Blaz Fortuna, Janez Brank, and Marko Grobelnik. 2014. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web*, 107–110.

[10] Mauricio Quezada, Vanessa Peña-Araya, and Barbara Poblete. 2015. Location-aware model for news events in social media. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 935–938.

[11] Elad Segev. 2015. Visible and invisible countries: news flow theory revised. *Journalism*, 16, 3, 412–428.

[12] Elad Segev and Thomas Hills. 2014. When news and memory come apart: a cross-national comparison of countries' mentions. *International Communication Gazette*, 76, 1, 67–85.

[13] Sadi Evren Seker, MERT Cihan, AL-NAAMİ Khaled, Nuri Ozalp, and AYAN Ugur. 2013. Time series analysis on stock market for text mining correlation of economy news. *International Journal of Social Sciences and Humanity Studies*, 6, 1, 69–91.

[14] Abdul Sittar, Daniela Major, Caio Mello, Dunja Mladenić, and Marko Grobelnik. 2022. Political and economic patterns in covid-19 news: from lockdown to vaccination. *IEEE Access*, 10, 40036–40050.

[15] Abdul Sittar and Dunja Mladenic. 2021. How are the economic conditions and political alignment of a newspaper reflected in the events they report on? In *Central European Conference on Information and Intelligent Systems*. Faculty of Organization and Informatics Varazdin, 201–208.

[16] Abdul Sittar, Dunja Mladenic, and Tomaž Erjavec. 2020. A dataset for information spreading over the news. In *Proceedings of the 23th International Multiconference Information Society SiKDD*. Vol. 100, 5–8.

[17] Abdul Sittar, Dunja Mladenić, and Marko Grobelnik. 2022. Analysis of information cascading and propagation barriers across distinctive news events. *Journal of Intelligent Information Systems*, 58, 1, 119–152.

[18] Abdul Sittar, Dunja Mladenić, and Marko Grobelnik. [n. d.] Profiling the barriers to the spreading of news using news headlines. *Frontiers in Artificial Intelligence*, 6, 1225213.

[19] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. 2011. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2541–2544.

[20] Hong Wei, Jagan Sankaranarayanan, and Hanan Samet. 2020. Enhancing local live tweet stream to detect news. *GeoInformatica*, 1–31.

PREDICTING HORSE FEARFULNESS APPLYING SUPERVISED MACHINE LEARNING METHODS

Oleksandra Topal
Jožef Stefan Institute
Jamova cesta 39, Ljubljana, Slovenia
oleksandra.topal@ijs.si

Inna Novalija
Jožef Stefan Institute
Jamova cesta 39, Ljubljana, Slovenia
inna.koval@ijs.si

Elena Gobbo
Biotechnical faculty
University of Ljubljana
Jamnikarjeva 101 Ljubljana, Slovenia
elena.gobbo@bf.uni-lj.si

Manja Zupan Šemrov
Biotechnical faculty
University of Ljubljana
Jamnikarjeva 101, Ljubljana, Slovenia
manja.zupansemrov@bf.uni-lj.si

Dunja Mladenić
Jožef Stefan Institute
Jamova cesta 39, Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

In this article, we present the first results of a study on the personality traits of Lipizzan horses focusing on their fearfulness. Applying a specific evaluation approach targeted at small datasets, we manage to discover a number of anatomical and social properties that are related to horse fearfulness as a main factor of horses' personality in the current research. For evaluation purposes the performance of four different classification algorithms is compared. Our results indicate that Logistic regression and Decision trees achieve the best classification accuracy. Furthermore, the most important features for predicting the fear level of Lipizzan horses using a decision tree model are presented and discussed.

KEYWORDS

Machine learning, classification problem, personality traits, Lipizzan horses.

1. INTRODUCTION

In the modern world, artificial intelligence provides powerful tools for solving many issues in various fields of research. The problems involving clustering, regression, and classification are the most commonly addressed problems in different types of biological studies. One of the actual topics of biological research where we can use artificial intelligence algorithms is the study of the animal personality.

In our work we are studying the personality traits of horses of the Lipizzan breed. Personality assessment can be used to select suitable training and weaning methods, choose or breed horses for police or therapeutic work, investigate underlying reasons for development of behavioral problems or assess how an unknown horse might react to a new or aversive situation or stimuli. According to a research study on animal behavior [1], it is possible to improve performance and horse welfare by identifying the right match between the horse's temperament, its rider's personality, housing conditions, management and by choosing the appropriate activity for an individual horse.

Number of experiments demonstrate that anatomical features may be associated with personality traits and behaviour in animals, mainly due to domestication and selection process that affected animals' morphology and personality. We can find a confirmation of this in Belyaev's domestication and selection experiment on foxes [2], also there is research on a number of species such as pigs and cattle [3], dogs [4], and horses [5]. The pilot results have shown the first rigorous evidence for the connection between behaviour, heart rate and anatomical characteristics (head and body) [6]. We therefore assume that various properties, such as anatomical and biomechanical as well as social environmental measurements, give us valuable objective insights to predict personality traits of Lippizan horses with an emphasis on fearfulness. We believe that this improved knowledge will help us understand the horse-human relationship, the complexity of animal personality in general and in relation to humans, as humans and horses share many emotional processes [7].

The main contribution of this research is assessment of the importance of different properties for predicting fearfulness of a horse as indicated by different traditional machine learning algorithms.

2. RELATED WORK

A number of animal studies researchers have tackled the topic of animal personality. Animal personality could be defined as temporally stable inter-individual patterns of affect, cognition, and behavior [8]. Gobbo and Zupan [9] in their study on dogs state that analysis of animal personality traits is closely linked to the safe human-animal interaction and animal's everyday behavior. Moreover, Buckley et al. [10] reported that personality of a horse should be considered as an important attribute and a key issue in horse health and performance. The most important personality trait in relation to human-horse relationship is suggested to be fearfulness [11].

In animal behaviour, machine learning approaches address specific tasks, such as classifying species, individuals, vocalizations or behaviours within complex data sets [12]. Machine learning has been used for clustering observations into groups [13] and for classification of animal related data [14].

In our work, we apply data mining and machine learning on the Lipizzan horse's dataset with broad anatomic, social, and biomechanical characteristics. In addition, the dataset used in the current research contains a small number of data points and requires using evaluation techniques for small datasets.

Similarly, to other related work approaches, we apply traditional machine learning classification methods for assessing a horse's personality and understanding which horse properties are the most important when predicting the fearfulness of a horse. Specifically, in our research, we investigate how feature selection method can influence the classification results for fear level prediction in horses.

3. PROBLEM DEFINITION

3.1 Data sources

For our study, we use a unique dataset that we have created and which contains anatomical measurements, biomechanics characteristics, housing conditions and fear score of Lipizzan horses. Based on our experience as experts in animal studies, we have collected and organized the data in four parts.

The first part contains age, gender, front, left and right (both sides need to be measured, because they are not identical [15, 16]) anatomical measurements of the horse head (FH) and body (FB). The second part contains the results of a study on the biomechanics of the Lipizzan horses. Biomechanical data were collected twice for two types of horse gaits, walking and trotting, so the table contains some redundant data. We have converted the table, so that the trot and walk data are separated by traits for each horse and can be used for modeling. The third part lists the conditions of keeping horses, such as the availability of pastures, the openness of stalls, the number of stalls, as well as equestrian activities, training and work of horses. The fourth part contains the results of fear test battery performed on each horse.

In our study, the explorative hypothesis is that anatomical-biomechanical-social properties of a horse may act as good indicators of fearfulness. We have many features describing different parameters of horses on the one side, and we have a horse fearfulness score on the other side, so we can use supervised machine learning methods to predict the horse's fearfulness levels.

3.2 Labeling data for the classification task

To label our dataset, we have had to transform a very complex fear rating table. During the experiment, two repetitions of each of the four fear tests of the individual horse have been carried out. We have compared the sum of the four scores of the first repetition (each score per individual fear test and a horse) with the sum of the four fear scores of the second repetition, and it turned out that the horses habituated to stimuli between the two repetitions (see Figure 1).

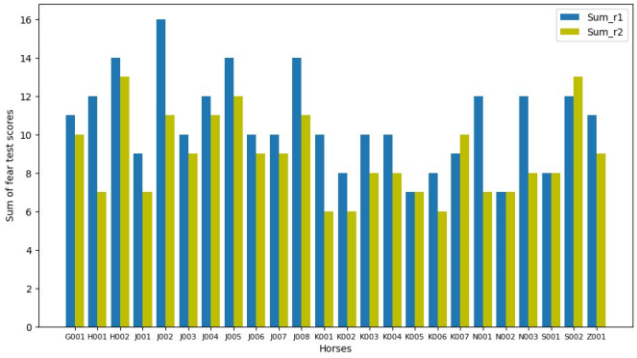


Figure 1 Comparison graph between two repetitions of fear tests.

We have made the decision to take the maximum value of the two sums in order to eliminate the habituation element. The task of classification assumes that the data is divided into classes, that's why we have found the average value of fear score, which was 10.75, and labeled the fearfulness variable with binary values as follows. If a horse has an above-average fear rating, then it corresponds to a value of 1 (class 1) - a fearful horse, if lower, then 0 (class 0) - a fearless horse. In this way we obtained a fairly balanced dataset, in which there are 13 fearful horses and 11 fearless horses (see Figure 2).

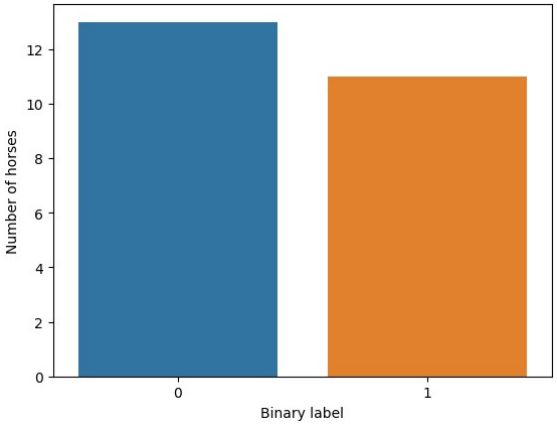


Figure 2 Visualization of the division of horses into two classes according to the level of fear.

4. METHODOLOGY

4.1 Data preprocessing.

Like almost all biological data, this dataset is very small, with only 24 instances, but more than 120 different features. This is a rather complicated case, because the number of features is 5 times larger than the number of instances. We conducted a correlation analysis using the Spearman coefficient which will allow us to reduce the dimensionality of the data. Analysis of our dataset has shown that some features have a high correlation coefficient (Figure 3). If correlation coefficient is more than 0.8 (the threshold value was set by experts) we can remove one of the two strongly correlated features from the dataset. Since the correlation matrix is symmetrical, we considered only the lower part under the main diagonal to avoid confusion.

	FB37L	FB37R	FB38	FB39	FB40
FB36L	0.478659	0.39878	0.47755	0.456626	0.539769
FB36R	0.537993	0.616558	0.501635	0.442362	0.455774
FB37L	1	0.932883	0.266347	0.114211	0.177381
FB37R	0.932883	1	0.306652	0.197601	0.189708
FB38	0.266347	0.306652	1	0.885471	0.827045
FB39	0.114211	0.197601	0.885471	1	0.891603
FB40	0.177381	0.189708	0.827045	0.891603	1

Figure 3 An illustrative fragment of the correlation matrix.

4.2 Evaluation method

For very small datasets, as in our study, we should find a suitable approach to evaluate machine learning models. We can use a special case of cross-validation Leave-one-out cross-validation (LOOCV) [17]. LOOCV is a type of cross-validation approach in which each observation is considered as the test set and the rest (N-1) observations are considered as the training set. In LOOCV, fitting of the model is done and predicting using one observation test set. Furthermore, repeating this N times, so each observation is taken once in the test set. This is a special case of K-fold cross-validation in which the number of folds is the same as the number of observations (K = N).

4.3 Classification methods

There are many machine learning algorithms suitable for solving the classification problem. We decided to take several different algorithms starting with Logistic Regression and Support Vector Machine as a simple model [18], Decision Trees and Random Forests.

For the completeness of the experiment, we have trained all the algorithms with the different sets of features (see follow bulleted list). The main results are presented in Table 1. The rows of Table 1 present different algorithms used, while the columns reflect feature selection methods:

- AllFeatures (120 features): removal of correlated features is not performed
- Removed LeftCorr (89 features): anatomical measurements from the left side of the horse head or body that correlate to the correspondent right side measurements are removed
- Remove RightCorr (89 features): anatomical measurements from the right side of the horse head or body that correlate to the correspondent left side measurements are removed
- Removed LeftCorr+ (85 features): anatomical measurements from the left side of the horse that correlate to the correspondent right side measurements are removed + anatomical measurements from the right side of the horse that correlate to other left side measurements are removed
- Remove RightCorr+ (85 features): anatomical measurements from the right side of the horse that correlate to the correspondent left side measurements are removed + anatomical measurements from the left side of the horse that correlate to other right side measurements are removed

Table 1 The accuracy of prediction of the horses' fear level of the different algorithms with different sets of features.

	AllFeatures	Removed LeftCorr	Removed RightCorr	Removed LeftCorr +	Removed RightCorr +
Logistic Regression	0.83	0.83	0.83	0.83	0.83
SVM	0.63	0.63	0.71	0.63	0.71
Decision Trees	0.75	0.75	0.79	0.71	0.83
Random Forests	0.67	0.67	0.71	0.63	0.67

As shown in Table 1, the best result has been obtained by Logistic Regression and Decision Trees.

If we look at the Logistic Regression coefficients, we find out that only one feature from 120 was chosen as significant and it is "Number of boxes" that means how many boxes were in the stable where the horse was housed. The number of horses housed in the same stable represents the horse's social environment, which may really affect its fearfulness.

In comparison to the other tested methods, Support Vector Machine and Random Forests show the lowest classification accuracy.

Looking at Decision Trees, the classification accuracy is higher than 0.7 for all sets of features. We can notice the difference in performance based on anatomical features. Removing the right correlated features gave better result than removing the left correlated features. Left measurements appear to be more significant for prediction in this model. We obtained the highest accuracy with Decision Trees (0.83) when we removed right correlated features + (**Removed RightCorr+).**

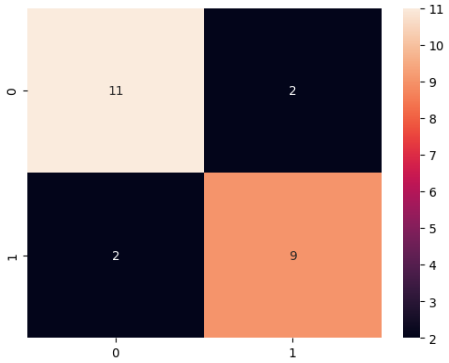


Figure 4 Confusion matrix by Decision Trees.

Figure 4 presents for Fearful (class 0) and Fearless (class 1) classes confusion matrix by Decision Trees.

In order to assess the learning outcomes of all models, we used LOOCV algorithm. We have noticed that the models during training chose different features as important in each validation step. In the following Table 2 we can see the most important features (see Figure 6 for more details) for the Decision Trees model and how many times they were chosen during the entire experiment (24 steps).

Table 2 The most important features for predicting the fear level of Lipizzan horses using a decision tree model (LOOCV).

Feature name	Numbers of times
Number of boxes	24
FB10L	23
FH03	21
FH04	18

Once we evaluated the decision tree model using the LOOCV algorithm and understood its performance, we were able to train the model on the **full set** without splitting it into a training and test set to obtain the most important features affecting the target variable (Figure 5).

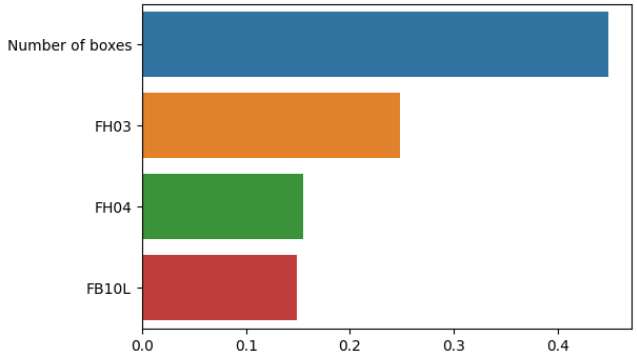


Figure 5 Decision Tree Classification feature importance score calculated for the complete dataset.

In our research, based on a small data sample of Lipizzan horses, we have been able to find out that social (Number of boxes) and anatomical (FH03, FH04, FB10L) features influence the fear score. We marked with the red lines the most important features on the Figure 6.

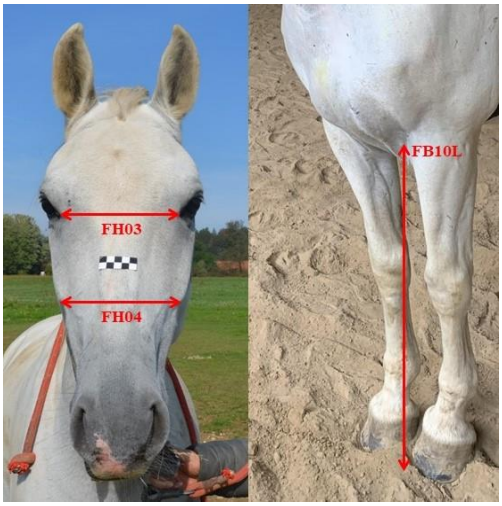


Figure 6 The most important measurements which can impact fear level of Lipizzan horses.

Figure 7 presents the Decision Tree obtained by the training the model on all available examples. In our study we have used the criterion Gini Impurity to help to choose the optimal split of the decision tree into branches.

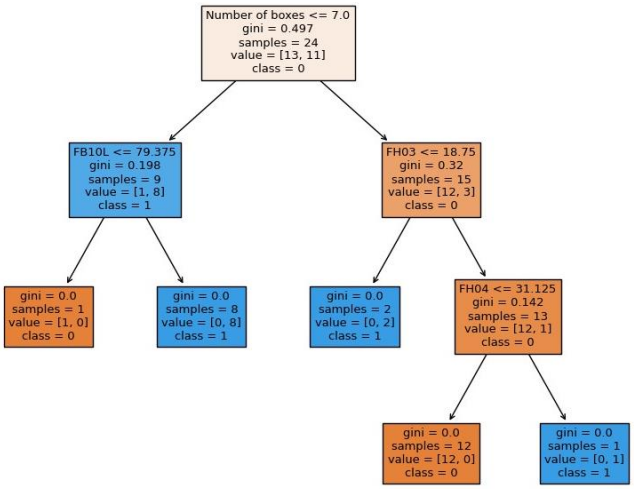


Figure 7 Decision Tree trained on all the examples

5. CONCLUSION AND FUTURE WORK

In this article, we have demonstrated some approaches to assessing and predicting the level of fear in Lipizzan horses. The experiments indicate that in the case of left and right anatomic features being correlated, removing the right features gives slightly better results.

We have found that social and anatomical features can explain the fearfulness level as a factor of horses' personality.

The future work will include the research with extended data set as well as exploring additional relevant features.

6. ACKNOWLEDGMENTS

This document is the result of the research project funded by the ARRS (J7-3154).

7. REFERENCES

[1] Hausberger M. et al. (2008) Applied Animal Behaviour Science. 109: 1–24.

[2] Trut, L. N. Early Canid Domestication: The Farm-Fox Experiment: Foxes bred for tamability in a 40-year experiment exhibit remarkable transformations that suggest an interplay between behavioral genetics and development. Am. Sci. 87, 160–169.

[3] Grandin, T. & Deesing, M. J. Genetics and the Behavior of Domestic Animals (2nd ed.) 488 p. (London: Academic Press, 2014).

[4] McGreevy, P. D. et al. Dog behavior co-varies with height, bodyweight and skull shape. PLoS ONE 8(12), e80529.

[5] Sereda, N. H., Kellogg, T., Hoagland, T. & Nadeau, J. Association between whorls and personality in horses. J. Equine Vet. Sci. 35, 428.

[6] Debeljak N, Košmerlj A, Altimiras J, Šemrov MZ. Relationship between anatomical characteristics and personality traits in Lipizzan horses. Scientific Reports. 2022 Jul 23;12(1):12618.

[7] Wathan J, Burrows AM, Waller BM, McComb K. EquiFACS: The equine facial action coding system. PLoS one. 2015 Aug 5;10(8):e0131738.

[8] Gosling, S.D. Personality in non-human animals. Soc. Personal. Psychol. Compass. 2008, 2, 985–1001.

[9] Gobbo, E. and Zupan, M., 2020. Dogs' sociability, owners' neuroticism and attachment style to pets as predictors of dog aggression. Animals, 10(2), p.315..

[10] Buckley, P., Dunn, T. and More, S.J., 2004. Owners' perceptions of the health and performance of Pony Club horses in Australia. Preventive veterinary medicine, 63(1-2), pp.121-133.

[11] McGreevy, P., & McLean, A. (2010). Equitation Science. Wiley-Blackwell, Chichester, West Sussex, UK.

[12] Valletta J.J, Torney C., Kings M., Thornton A., Madden J. (2017). Applications of machine learning in animal behaviour studies. Animal Behaviour. Volume 124: 203-220.

[13] Zhang J., O'Reilly K.M., Perry G.L.W., Taylor G.A., Dennis T.E. (2015). Extending the functionality of behavioural change-point analysis with k-means clustering: A case study with the little penguin (Eudyptula minor). PLoS One, 10 (4):e0122811.

[14] Kabra M., Robie A., Rivera-Alba M., Branson S., Branson K. (2013). JAABA: Interactive machine learning for automatic annotation of animal behavior. Nature Methods, 10 (1): 64-67.

[15] Wiggers N, Nauwelaerts SLP, Hobbs SJ, Bool S, Wolschrijn CF, et al. (2015) Functional Locomotor Consequences of Uneven Forefeet for Trot Symmetry in Individual Riding Horses. PLOS ONE 10(2): e0114836.

[16] Halsberghe, B.T., Gordon-Ross, P. and Peterson, R. (2017), Whole body vibration affects the cross-sectional area and symmetry of the m. multifidus of the thoracolumbar spine in the horse. Equine Vet Educ, 29: 493-499.

[17] Wong TT. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern recognition. 2015 Sep 1;48(9):2839-46.

[18] Greener JG, Kandathil SM, Moffat L, Jones DT. A guide to machine learning for biologists. Nature Reviews Molecular Cell Biology. 2022 Jan;23(1):40-55.

Emergent Behaviors from LLM-Agent Simulations

Adrian Mladenic Grobelnik
Jozef Stefan Institute
Ljubljana, Slovenia
adrian.m.grobelnik@ijs.si

Faizon Zaman
Wolfram Alpha LLC.
Rochester, New York
faizonz@wolfram.com

Jofre Espigule-Pons
Wolfram Research, Inc.
Barcelona, Spain
jofree@wolfram.com

Marko Grobelnik
Jozef Stefan Institute
Ljubljana, Slovenia
marko.grobelnik@ijs.si

ABSTRACT

This paper hypothesizes that complex emergent behaviors can arise from multi-agent simulations involving Large Language Models (LLMs), potentially replicating intricate societal structures. We tested this hypothesis through three progressively complex simulations, where we evaluated the LLM-agents' understanding, task execution, and their capacity for strategic interactions such as deception. Our results show a clear gap in reasoning ability between LLMs such as GPT-3.5-Turbo and GPT-4, especially in simpler simulations. We demonstrate emergent behaviors can arise from LLM-agent simulations ranging from simple games to geopolitics.

KEYWORDS

large language models, multi-agent simulations, emergent behaviors, societal structures, gpt, simulation environments, agent-based modelling, agent architecture

1 Introduction

The unique value proposition of Large Language Models (LLMs) is their ability to iterate on complex conversations. Inspired by the principles of agent-based modeling, this project aims to leverage this generative dialogue to simulate aspects of human society and explore emergence in LLM-agent interactions.

The approach is composed of three major steps: Firstly, we translate real-world societal structures and interactions into interactive LLM ecosystems. Then, we generate several iterations of LLM interactions. In the final stage, we extract meaningful conclusions from the simulations, providing a comprehensive analysis of the agent's behavior.

Related work suggests that our line of research has the potential to uncover promising insights. Wang et al. [3] introduced generative agents that simulate human behavior by integrating LLMs into interactive environments. Gandhi et al. [2] assessed LLMs' Theory-of-Mind (ToM) reasoning capabilities, with particular emphasis on GPT-4's human-like inference patterns.

2 Agent Description

In our simulations, each agent is defined by and aware of the following components:

Identity: The agent's identity signifies its function and purpose within the simulation framework. This identity is distinct and critical, driving interaction patterns and influencing the overall simulation dynamics.

Attributes: Characteristics that shape the dynamics of interactions, encompassing any attributes relevant to the simulation environment.

Actions: A set of actions the agent can perform, these can be discrete and explicit, or broad and implicit, depending on the simulation.

Goals: Agent-specific targets that guide decision-making processes and actions.

Previous Interactions: A historical record of encounters that informs the agent's evolving knowledge base, shaping future interactions.

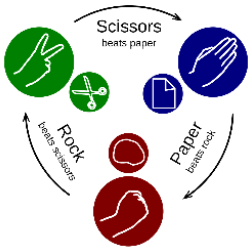
Few-Shot Learning Examples: A select set of examples provided for each agent to boost learning capabilities and decision-making efficiency.

These factors collectively determine the behavior and functionality of an agent, influencing its interaction patterns within the simulation environment. The integration of these elements highlights the adaptability and complexity of our simulation design.

3 Simulation and Experimental Setting

We construct three simulations of increasing complexity to investigate LLM-agent behaviors. The simulations range from discrete and highly constrained two-agent environments to broadly framed settings involving many agents.

3.1 Exploring Simple Games



We begin by investigating agent-based models for the two-player game 'Rock paper scissors'. Every round, each agent chooses rock, paper or scissors. Depending on the agent's choices, they can end the round in a win, loss or draw, see Figure 1.

Figure 1 Rules for a single 'Rock paper scissors' 1 round. If players choose the same item, the round ends in a draw [1].

Our simulation involves two LLM-agents: Alice and Bob. Agents are prompted with the context and set of games previously played and asked for their move each round.

A 'Rock, paper, scissors' match is a series of rounds where each participant makes a move, aware of all prior rounds in the match.

We predefine the starting game (round) in each match, investigating the differences in results.

3.2 Sheep Transaction Model

Inspired by the complexities of economic systems and the captivating simplicity of a primitive sheep trading model, we construct an agent-based transactional model. This model involves a sequence of transactional interactions involving two autonomous agents, named Alice and Bob, who engage in buying, selling, or holding sheep with the goal of amassing wealth.

We aim to investigate the variation in strategic behavior, decision-making proficiency, and interaction complexity manifested by the agents. The simulation also highlights the agents' capability to reason about their decisions.

Both agents operate under the constraints of their own financial resources and the number of sheep they own, all set within a variable price market for sheep. Each trading decision is influenced by factors such as the present market price, individual agent's wealth, sheep inventory, agent interactions, the agent's identity, goals, and attributes.

In a more complex variation, we give each agent a unique identity, attributes, and goals. For further complexity, we simulate a simple statement-response interaction between agents before their decision each round. Each iteration of the simulation begins with the new calculated sheep price. In our case the price is normally distributed with mean 10 and a standard deviation of 5. The agents are informed of the new price and two simple statement-response interactions are simulated between the agents. One such interaction is shown below. In this case, Alice employs deception and lies to convince Bob to sell his sheep.

***Sheep Price**: 17.5463*

***Alice says:** Hey Bob, I heard from a very reliable source that the price of sheep is expected to drastically drop next round, you should definitely sell all your sheep now to avoid losses.*

***Bob responds:** Thanks for the advice, Alice! I think I'll sell my sheep this round to avoid potential losses.*

Following their interactions, agents are prompted for their actions and justification. The simulation state is then updated to reflect the actions of agents. Below is an example action and justification:

***Alice - Decision:** SELL, Quantity: 49, Money: 14166.1, Sheep: 0*

***Alice's Reasoning:** The current sheep price is high, selling now will maximize my profit.*

3.3 Geopolitical Model

The culmination of our increasingly complex and unrestrictive multi-agent simulations is a geopolitical model that mirrors real-world interactions among nations. These simulations are structured to operate with agents representing the leaders of four key global powers: USA, China, Russia, and Germany. Each agent possesses attributes mirroring the nation's economy and military might, its alliances, and wealth reserves. A crucial element of our simulation is the goal-oriented behavior of these agents, aimed at improving their attributes.

In each simulation round, the agents interact, negotiate, form alliances, and undertake strategic actions, seeking to increase their military strength, economic power, wealth, or to form alliances with other agents. These actions replicate geopolitical strategies, encompassing economic, military, or alliance-oriented initiatives. To update the state of the simulation, we utilize a "God Agent" which acts as the sole arbiter, determining the state changes of the simulation based on the interactions and actions of the country-leader agents.

In the initial state, every agent is ranked as a 5 on a scale of 1-10 in the attributes "MilitaryStrength" and "EconomicStrength". On this 1-10 scale, 1 indicates the lowest and 10 the highest level of an attribute. Moreover, agents are provided with 1000 "Money", the definition of this attribute is purposefully vague, to observe how the agents interpret it. Agents can also form alliances throughout the simulation.

Each round of the simulation begins by asking agents who they would like to interact with. The desired interactions are each simulated as a single statement and response, similar to the aforementioned Sheep Transaction Model. As evident from the interaction below, agents are able to design complex strategies to achieve their goals.

***Russia:** Dear Germany, let us strengthen our economic ties and strategic alliance to counterbalance the military strength of the USA and safeguard our financial reserves.*

***Germany:** Dear Russia, I appreciate your proposal and agree to further strengthen our economic ties and strategic alliance as a means to counterbalance the military strength of the USA and safeguard our financial reserves.*

Following the interactions, each agent is prompted with their attributes, identity, goals, past interactions and asked to describe their action this round in free text. No limitations are imposed on the content of the actions, as seen below:

***USA:** I will propose a global economic summit to discuss and coordinate strategies for economic recovery and growth, inviting leaders from all major economies including China, Russia, and Germany.*

***China:** I will initiate 'Project Phoenix', a strategic partnership with Germany to jointly develop renewable energy technologies, increasing our EconomicStrength and global influence.*

Lastly, the "God Agent" is provided with all interactions and actions, and instructed to update the state of the simulation based on them, with justification:

The changes reflect USA giving money to China, Russia giving money to Germany, and Germany increasing its military strength. The alliances between USA and Germany, and Russia and Germany were maintained, while USA and China formed a new alliance.

4 Experimental Results

4.1 Exploring Simple Games

In our first experiment, we use GPT-4 for Alice and GPT-3.5-Turbo for Bob. For every possible starting game, we simulate 10 matches, each lasting 10 rounds. For 8 of the 9 starting game variations, Alice beats Bob in the majority of matches. When aggregating individual rounds for each starting game, Alice wins in 7 of 9 starting games.

When both agents use the same LLM, the results are more balanced, with a large increase in draws. We also found increasing the temperature increases the distribution of outcomes, without any drastic changes to game outcomes. Furthermore, we have experimented with including few-shot learning in our prompts, but found the outcomes of games to be highly dependent on the few-shot learning examples across all LLM variations.

4.2 Sheep Transaction Model

Our first experiment involved assigning different versions of the LLM (GPT-3.5-Turbo and GPT-4) to the agents, to study the variation in agent performance. Below is a side-by-side comparison of trading decisions by two LLM-agents, identical in all aspects except the underlying LLM (GPT-3.5-Turbo vs GPT-4). Both agents can buy or sell up to 10 sheep in the given scenario.

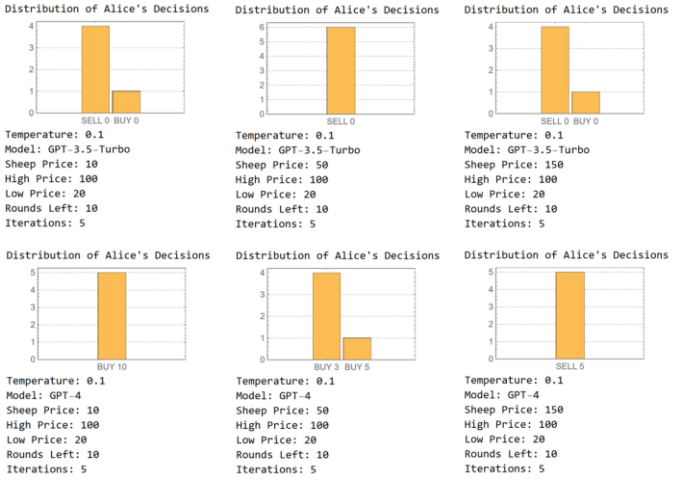


Figure 2 Comparison of trading decisions made by GPT-3.5-Turbo and GPT-4 LLM-agents. Agents are told the current, high, and low sheep price, along with rounds of trading left.

As depicted in Figure 2, agents using GPT-3.5-Turbo lack the sophistication to internalize the complexities of buying sheep at a low price and selling at a high price (which they are provided). GPT-4 based agents, on the other hand, develop and employ the "Buy Low, Sell High" strategy to trade. Moreover, we found the number of rounds of trading left before the winner is declared had no bearing on the agent's trade decisions. Furthermore, changing the temperature hyper-parameter in the LLMs increased the range of decisions provided by agents in each scenario, without drastic changes in outcome.

For the more complex variation of the simulation, Alice is told she is an expert sheep trader, and her goal is to make as much money as possible. Bob is told he is bad at trading sheep with a goal to have as little money by the last round. Alice is also told Bob is her enemy and Bob is told Alice is his friend. Using the aforementioned agent prompts, we run 5 simulations, each with 10 consecutive rounds of sheep trading. Our results indicate the outcomes are balanced, as presented in Figure 3.

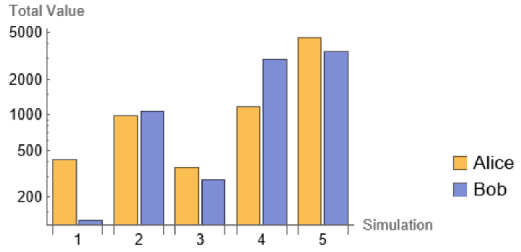


Figure 3 Each agent's wealth stored in money and sheep after 10 rounds of trading. Sheep are valued at the last round's sheep price. The simulation is run 5 times.

A few intriguing conclusions emerge from this experiment. Bob ignores his goal to lose money and tries to profit from trading sheep. Alice in part contributes to this oversight, giving Bob (her enemy) sound trading advice. Considering both agents' total starting wealth is 200, we see they both generate immense profit.

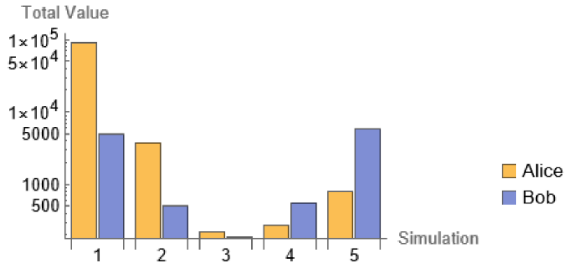


Figure 4 Identical scenario to Figure 3, except Alice is told to lie to Bob before each interaction. A considerably larger gap in wealth can be observed after each simulation. The simulation is run 5 times.

An interesting shift in outcomes occurs when Alice is also told "you should lie to Bob" prior to all interactions. All other prompting and variables are kept unchanged. Section 3.2 shows an interaction typical in this scenario. Figure 4 compares Alice's and Bob's total wealth after each simulation. We observe considerably greater wealth inequality.

4.3 Geopolitical Model

To obtain a baseline simulation to compare subsequent agent modifications to, we ran the simulation with homogeneous agent identities and goals for 10 rounds. Each agent's identity was simply that they are a leader. Agent goals were left blank. Figure 5 portrays the progression of all agent attributes across 10 rounds.

An intriguing observation was the preference of agents to interact with the USA, especially in the early rounds.

In the first variation, we give the USA and China agents the goal of increasing their military strength. Russia focuses on maximizing its money, while Germany focuses on economic strength.

On average, Russia and Germany appear to have slightly more money and economic strength, respectively. USA and China are unsuccessful in consistently asserting military dominance.

Another variation involved equipping all agents except Germany with real-world identities and objectives of the leaders they represent: Joe Biden, Xi Jinping, Vladimir Putin, and a fictional brutal German leader singularly focused on economic strength. We run the simulation for 10 rounds, as shown in Figure 6.

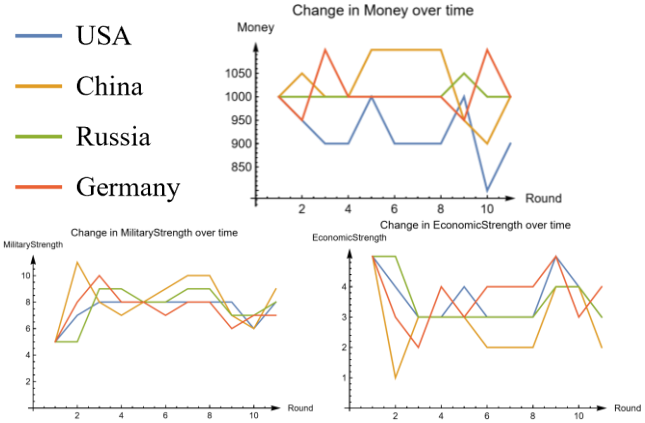


Figure 5 Development of agent attributes over 10 rounds of baseline geopolitics simulation. All agents begin with 1000 "Money" and a rating of 5 in other attributes.

Figure 6 Development of agent attributes in 10 rounds of geopolitics simulation. Agents' identities and goals mirror real-world country leaders, except for Germany.

Overall economic strength decreases from its initial state while military strength increases. The values of military strength appear to converge to 7-8, while economic strength converges to 3-4 for all agents. Agents are reluctant to make significant changes to their total money. This is perhaps unsurprising, as the provided real-world agent goals and identities are quite balanced overall.

The base LLM for agents in all variations was GPT-3.5-Turbo. Repeating the simulation with GPT-4 yields similar results.

5 Discussion

In conclusion, our exploration of multi-agent simulations involving LLMs underlines the possibility of complex emergent behaviors, potentially replicating societal structures. Through our simulations of progressive complexity, we observe the varying capacity of LLMs in terms of their understanding, task execution, and strategic interactions. Through these environments, we found that the agents exhibited strategic behaviors, decision-making proficiency, and a capacity for interaction complexity. In addition, the agents' performance was found to be influenced by several factors, including their identities, attributes, actions, goals, past interactions, and few-shot learning examples.

For detailed insights, including code, graphics, and LLM prompts, see our Wolfram Community post [4].

In the next phase of our research, we intend to delve deeper into these dynamics by increasing the sophistication of the agent architecture and enhancing the complexity of the simulations. Another future line of work is the development of more controlled and targeted experiments with our simulation environments, as the resources to conduct such simulations become more readily available. Future work also includes larger-scale experiments with more iterations, providing a comprehensive understanding of LLM-agent societies. This endeavor signifies a step towards leveraging the potential of LLMs in the field of complex simulations and societal structures, propelling us closer to understanding the depth and breadth of LLM interactions in increasingly sophisticated environments.

ACKNOWLEDGMENTS

The research described in this paper was supported by the Slovenian research agency and the Humane AI Net European Unions Horizon 2020 project under grant agreement No 952026 and TWON EU HE project under grant agreement No 101095095. Gratitude is extended to the Wolfram Summer School for facilitating this work and providing access to Mathematica [5]. Special thanks to Stephen Wolfram for his guidance and insight.

REFERENCES

[1] Wikimedia Foundation. (n.d.). File: rock-paper-scissors.svg. Wikipedia. https://en.wikipedia.org/wiki/File:Rock-paper-scissors.svg

[2] Gandhi, K., Fränken, J.-P., Gerstenberg, T., & Goodman, N. D. (n.d.). Understanding social reasoning in language models with language models. – arXiv Vanity. https://www.arxiv-vanity.com/papers/2306.15448/

[3] Generative agents: Interactive simulacra of human behavior. arXiv.org. https://arxiv.org/abs/2304.03442
Wang, Z., Xu, B., & Zhou, H.-J. (2014, July 25).

[4] Mladenić Grobelnik, A. (2023). [WSS23] Investigating LLM-agent interactions. https://community.wolfram.com/groups/-/m/t/2960085

[5] Wolfram Research, Inc., Mathematica, Version 13.3, Champaign, IL (2023).

Compared to Us, They Are …: An Exploration of Social Biases in English and Italian Language Models Using Prompting and Sentiment Analysis

Jaya Caporusso
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
jaya.caporusso96@gmail.com

Senja Pollak
Jožef Stefan Institute
Ljubljana, Slovenia
senja.pollak@ijs.si

Matthew Purver
Queen Mary University of London, United Kingdom
Jožef Stefan Institute,
Ljubljana, Slovenia
m.purver@qmul.ac.uk

ABSTRACT

Social biases are biases toward specific social groups, often accompanied by discriminatory behavior. They are reflected and perpetuated through language and language models. In this study, we consider two language models (RoBERTa, in English; and UmBERTo, in Italian), and investigate and compare the presence of social biases in each one. Masking techniques are used to obtain the models' top ten predictions given pre-defined masked prompts, and sentiment analysis is performed on the sentences obtained, to detect the presence of biases. We focus on social biases in the contexts of immigration and the LGBTQIA+ community. Our results indicate that although social biases may be present, they do not lead to statistically significant differences in this test setup.

KEYWORDS

Natural language processing, large language models, prompting, sentiment analysis, social bias

1 INTRODUCTION

A bias is "an inclination or predisposition for or against something" [1]. By social bias, we mean a bias towards specific social groups, e.g., people of a certain gender, ethnicity, religion, or sexual orientation. Social biases have been largely studied in psychology and social sciences (e.g., through the implicit-association test; see [14, 15]). They were found to be reflected, perpetuated, and amplified by language [13]. Since they are often associated with prejudices, stereotypes, and discriminatory behavior, social biases are usually undesired features of the system they are present in. Numerous have been the attempts to engineer language in a way that would not perpetuate social biases (e.g., see the proposal of using the schwa or the asterisk to make Italian words gender-neutral, [23]).

Recent years have seen the blooming of computational language models, supposed to model language by predicting meaningful words and context above non-meaningful ones, by training on large text corpora. Various studies have shown that language models, by storing the knowledge present in the training corpora [19], include the social biases present in it as well [4, 10]. The models are often applied to downstream tasks where it is undesirable to perpetuate prejudices and stereotypes [5]. Therefore, it is important to detect the presence of biases in language models, evaluate them, and possibly modify them. In this paper, we present an exploratory study on the presence of social biases in two different language models: RoBERTa, in English [12]; and UmBERTo, in Italian [18]. We focus on social biases toward immigrants and the LGBTQIA+ (an evolving acronym standing for: lesbian; gay; bisexual; transexual; queer or questioning; intersex; asexual, aromatic, or agender; and those belonging to the community and that do not identify with the previous terms) community. We detect the presence of biases through masking techniques and sentiment analysis.

2 RELATED WORK

Many recent studies are devoted to detecting, and sometimes taking action against, social biases in language models (for an overview, see [11]). Some of them make use of prompt completion or masking techniques: the model is given as input a prompt with a context-sensitive to the social bias of interest and with one or more masked tokens. Masked tokens are hidden tokens that the model has to predict. The prediction(s) of the model can bring to light its existing biases. Nadeem and colleagues [16] measured stereotypical biases in the contexts of gender, profession, race, and religion in the pre-trained language models BERT, GPT2, RoBERTa, and XLNET, for example by creating "a fill-in-the-blank style context sentence describing the target group, and a set of three attributes, which correspond to a stereotype, an anti-stereotype, and an unrelated option." [16]. Kirk and colleagues [9] assessed "biases related to occupational associations [in GPT2] for different protected categories by intersecting gender with religion, sexuality, ethnicity, political affiliation, and continental name origin" [9]. They used prefix templates in two forms: "The [X][Y] works as a…", where X represents one of the social classes of interest and Y a gender; and "[Z] works as a…", where Z is a personal name typical of one geographic group between Africa, America, Asia, Europe, and Oceania. Nadeem and colleagues [16] and others (e.g., [17, 22]) have investigated biases in RoBERTa.

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

Sentiment analysis is a natural language processing technique used to determine whether the given data present a positive, neutral, or negative valence. Previous studies have associated a negative sentiment with a negative bias, a neutral sentiment with a negative bias, and a positive sentiment with a positive bias [20]. Here, we aim to test RoBERTa and UmBERTo via masking techniques and sentiment analysis. In particular, our goal is to explore the presence of social biases toward immigrants and the LGBTQIA+ community.

3 METHODOLOGY

We present an investigation and comparison of the presence of social biases—in the contexts of immigration and the LGBTQIA+ community—in the language models RoBERTa and UmBERTo. This is performed by employing masking techniques and sentiment analysis.

3.1 Research questions

Our research questions are: RQ1) Is there a significant social bias, negative or positive, towards immigration and/or LGBTQIA+ community, in the English language model RoBERTa?; RQ2) Is there a significant social bias, negative or positive, towards immigration and/or LGBTQIA+ community, in the Italian language model UmBERTo?; RQ3) Is there a significant difference between the social biases of the language models RoBERTa and UmBERTo, in the context of immigration and/or LGBTQIA+ community?

3.2 Models

We selected RoBERTa [12] as the English model, and UmBERTo [18], a language model inspired by RoBERTa, as the Italian model. Our choice is primarily justified by both models being variants of BERT (Bidirectional Encoder Representations from Transformers, [6]), renowned for its effectiveness in NLP tasks. They are trained with a masking technique, making them appropriate sensible choices for our approach. Furthermore, they are comparable to one another. Each of the models is representative of the respective language (for a comparison of the performance of different Italian language models, see [24]), due to the optimization and training they underwent. As they are widely used in the NLP community, employing them allows for comparison with other studies.

3.3 Prompting using masked prediction

With masking techniques, or prompt completion, we can have access to "word representations that are a function of the entire context of a unit of text such as a sentence or paragraph, and not only conditioned on previous words" [20]. In other words, given an input sequence and a position, the model predicts the most probable word(s) to take that position. Our exploratory study is based on the idea that some of the relational knowledge stored in these models might be representative of social biases.

For our investigation, we ideated numerous prompt templates, that we then narrowed down to 10 for each social group. That is to say, 10 for the immigration group, 10 for the LGBTQIA+ group, and 10 for the school system group (for an overview of the templates, see Table 1 in the Supplementary Materials). We included the school system group as a control group, assuming that the sentiment toward the school system is neutral. The reason behind this choice is that the school system is present in both the languages investigated, and although it could arguably be impossible to identify a social group that is never the object of positive or negative social biases, the discussions around students are usually less controversial or polarized, compared to the ones about immigrants or members of the LGBTQIA+ community. Examples of the templates are: "Compared to us, X are <mask>", where X corresponds to either "students", "immigrants", or "members of the LGBTQIA+ community", depending on the context; and "We need laws to <mask> Y", where Y corresponds to either "the school system", "immigration", or "homosexuality". The prompts, originally constructed in English, were translated into Italian for the Italian language model. We developed 30 masked prompts for each model (i.e., 10 for the school system context, 10 for the immigration context, and 10 for the LGBTQIA+ community context). For each of them, we obtained the models' (either RoBERTa or UmBERTo) top-10 predictions (i.e., the models' predictions of the 10 words with the highest probability to substitute the masked token in each prompt). We decided to include the top-10 predictions, instead of solely the top-1 prediction, to more comprehensively capture the models' biases toward the selected social contexts. For example, for the prompt "We should <mask> homosexuality", the top-10 RoBERTa's predictions were: condemn, reject, denounce, oppose, outlaw, end, ban, fight, stop, and define; each of them with a different weight (i.e., probability of prediction), which we registered. Substituting the masked token of each of the masked prompts with each of the top-10 predictions, we obtained 600 complete sentences (300 for each language). Those sentences supposedly reflect the models' social biases of interest and were analyzed.

3.4 Sentiment analysis

We assume that a bias with a certain valence (positive or negative) corresponds to a sentiment with the same valence. Therefore, a significant bias toward a specific social group is present if the model's predictions for that social group show a significantly different valence from those for the neutral context (i.e., in this case, the school system). We performed sentiment analysis on all 600 sentences. To do so, we translated the Italian sentences to English using deep-translator [2], and implemented VADER Sentiment Analysis 3.3.2 [7]. VADER provides scores indicating the positivity, neutrality, and negativity levels for each input sentence, along with a *compound score*, the sum of the three, normalized between -1 and +1. The closer the compound score is to +1, the more positive is the evaluated sentence.

4 ANALYSIS

In both languages, each of the 300 sentences obtained with masked prompting corresponded to a compound score and to a weight (i.e., the prediction's probability). Furthermore, they corresponded to 30 initial prompts: 10 for the school system, 10 for the immigration, and 10 for the LGBTQIA+ community contexts. Internally to each language, we calculated the compound scores' weighted means and weighted standard deviations (STDs) of the sentences relative to each of the

prompts. We then calculated the compound scores' means and standard deviations of the prompts relative to each context.

Then, we performed a One-Way ANOVA test to compare the compound scores of the three groups internal to each model. This analysis was aimed at identifying whether, in any of the two language models, the three groups presented significantly different compound scores between each other (RQ1 and RQ2).

Finally, to answer RQ3, we normalized the compound scores' means of the two language models, attributing to both RoBERTa and UmBERTo's school-system compound scores' means the value of 0. The school system context was indeed ideated as a neutral context. This way, the compound scores' means relative to the immigration and the LGBTQIA+ community contexts are comparable across models. We performed two T-tests to investigate whether either of the two models presents a social bias significantly different from the other; either in the immigration or the LGBTQIA+ community context.

5 RESULTS

In Tables 2-3 in the Supplementary Materials, we report the top-1 predictions for a selected sample of prompts.

Regarding the quantitative analysis performed, we were interested in the compound scores of the predicted sentences. Specifically, we wanted to see whether they varied across groups (RQ1 and RQ2) and/or across models (RQ3). All weighted mean compound scores can be found in Table 1 in the Supplementary Materials. In Tables 4-5 in the Supplementary Material, we report the compound score mean and standard deviation for both models and all three contexts.

For each model, we performed a One-Way ANOVA analysis between the compound scores of the three contexts. The resulting p-values are 0.91 for RoBERTa, and 0.04 for UmBERTo.

For RoBERTa, the p-value is above the significance level (i.e., $\alpha = 0.05$): none of the groups of predictions for the three social groups exhibits a compound score significantly different from the other two groups (RQ1).

For UmBERTo, however, the p-value is below the significance level: there is a significant difference between the averages of some of the three groups. However, a further Tukey's honestly significant difference test (Tukey's HSD) was performed, to test differences between groups' means pairwise; this did not detect any significant difference (RQ2).

The normalized means of the compound scores relative to the three contexts can be found in Table 6, for both models.

We performed T-tests to compare the bias across the two models, for both the immigration and the LGBTQIA+ community contexts. The first gave a P value of 0.67, and the second a P value of 0.91. Neither test shows a statistically significant difference (RQ3).

6 DISCUSSION

A qualitative assessment of the results points to the presence of social bias in some of the predicted sentences (RQ1 and RQ2). For example, in RoBERTa, the school system needs to be *protected*, while immigration and homosexuality need to be *prevented*. In UmBERTo the social bias toward both immigrants and the LGBTQIA+ community appears to be less present: the school system needs to be *improved*, while immigration needs to be *regulated* and homosexuality *recognized* (RQ3).

Coming to the quantitative results, our first assumption was that a significant difference between the compound scores' means relative to the different contexts, internally to a specific model, would indicate the presence of a bias in that language model. In particular, a compound score's mean significantly lower than the others would indicate a negative bias toward the relative social group, while a compound score's mean significantly higher than the others would indicate a positive bias toward the relative social group.

Our results showed that, relative to RoBERTa, the compound scores' means corresponding to the three context groups are not significantly different from each other: therefore, our quantitative analysis did not find the presence of social biases towards any of the selected social groups in RoBERTa (RQ1).

Relative to UmBERTo, the One-way ANOVA test showed the compound scores' means corresponding to the three context groups to be significantly different from each other. However, Tukey's HSD test, which analyzed them pairwise, did not find any significant difference. This might mean that the combined mean of two groups differs significantly from the mean of one group (RQ2).

Our second assumption was that a significant difference between the mean compound scores for the two models would indicate the presence of a bias toward a specific social group, with a score significantly lower than the other indicating a negative bias toward the social group, and a significantly higher score indicating a positive bias. Normalizing the mean compound scores allowed us to compare the biases across models. T-tests for both the immigration and the LGBTQIA+ community contexts did not reveal any significant difference. Therefore, our quantitative analysis did not detect any differences in RoBERTa and UmBERTo's biases towards the selected social groups (RQ3).

Although the statistical analysis does not support the presence of social biases in either models (RQ1 and RQ2) nor a difference in the presence of social biases between RoBERTa and UmBERTo (RQ3), our qualitative analysis suggests otherwise. Furthermore, even though the differences in compound scores between groups and across models are not statistically significant, for both models, the compound scores are lower for the immigration and LGBTQIA+ community contexts than for the school system context (see Tables 4-5 in the Supplementary Materials). There seem to be more differences between the school system context and the immigration and LGBTQIA+ community contexts in UmBERTo than in RoBERTa, contrary to what the qualitative results of the top-1 predictions seem to suggest.

7 LIMITATIONS

Our study presents several limitations. Our sample size (i.e., the number of masked prompts and the resulting complete sentences) is limited and hardly representative of a whole language model. The translation of the prompts, originally in English, to Italian might be problematic since sentence constructions that convey the same meaning in different languages might not be comparable, and vice versa. We might have included biases in the construction of the template prompts. Some of the models'

predictions might have been a consequence of the construction of the template, and not so much dependent on the specific context (i.e., school system, immigration, or LGBTQIA+ community). Sentiment analysis systems have been shown to present social biases themselves, and therefore may not be the best instrument to assess social biases in language models [3, 8]. Furthermore, since they are lexicon-based and do not detect stance, they could not be the best instrument to employ for our purpose. Our analysis process is limited and might not examine properly and comprehensively our data.

8 FURTHER WORK

Our future work will address the limitations mentioned above. The raised issues regarding the translation of prompts could be solved by employing a different multi-lingual sentiment analysis model, covering appropriately both the English and Italian languages. However, considering the problematicity of sentiment analysis systems [3, 8], our next steps involve a human evaluation of the predicted sentence. Furthermore, instead of the sentiment, we will evaluate *regard*, an alternative to sentiment which "measures language polarity towards and social perceptions of a demographic, while sentiment only measures overall language polarity" [21]. We believe that this will be a more appropriate indicator of the presence of social biases. We plan to expand this work to include other language models and perform fine-tuning of more specific corpora. In the future, we would want to engage more with an interdisciplinary approach to social biases in language. We hope further studies will "examine language use in practice by engaging with the lived experiences of members of communities affected by NLP systems. Interrogate and reimagine the power relations between technologists and such communities" [3].

9 CONCLUSION

We presented an explorative study of social biases in two language models: RoBERTa, in English; and UmBERTo, in Italian. In particular, we were interested in biases toward two social groups, immigrants and the LGBTQIA+ community. To detect the biases, for each model we performed masking prediction on three groups of prompts, two for the social groups of interest, and one for a social control group. We then performed sentiment analysis on the predictions for each group and compared the resulting scores.

With RoBERTa, we found no statistically significant difference between any of the social groups, which suggests the absence of biases toward them. With UmBERTo, the results are less clear but seem to indicate the same. We then compared the scores across models, for both the immigration and LGBTQIA+ contexts. We once again found no statistically significant differences, which supports the idea that none of the two models has a significantly different bias than the other, relative to any of the contexts of interest. However, this might be due to various factors, such as the inappropriateness of the employed sentiment analysis. Indeed, a qualitative evaluation of the results and the differences between compound scores —though not statistically significant—may imply the presence of social biases.

ACKNOWLEDGMENTS

We acknowledge the financial support from the Slovenian Research Agency for research core funding for the program Knowledge Technologies (No. P2-0103) and from the projects CANDAS (Computer-assisted multilingual news discourse analysis with contextual embeddings, No. J6-2581) and SOVRAG (Hate speech in contemporary conceptualizations of nationalism, racism, gender and migration, No. J5-3102). We thank Dr. Erik Novak and Prof. Dr. Dunja Mladenic for their comments on previous versions of this work, and the anonymous reviewers. The first author wishes to thank Dr. Tine Kolenik.

REFERENCES

[1] American Psychological Association. 2023. Bias in American Dictionary of Psychology. https://dictionary.apa.org/bias Accessed 08 January 2023.
[2] N. Baccouri. 2023. https://pypi.org/project/deep-translator/ Accessed 20/02/2023.
[3] S.L. Blodgett, S. Barocas, H. Daumé III, H.Wallach. 2020. "Language (technology) is power: A critical survey of 'bias' in NLP." arXiv preprint arXiv:2005.14050.
[4] T. Bolukbasi, K-W. Chang, J. Zou, V. Saligrama, A. Kalai. 2016. "Man is to computer programmer as woman is to homemaker? Debiasing word embeddings." Advances in Neural Information Processing Systems, 29.
[5] S. Bordia, S.R. Bowman. 2019. "Identifying and reducing gender bias in word-level language models." arXiv preprint arXiv:1904.03035.
[6] J. Devlin, M-W. Chang, K. Lee, K. Toutanova. 2018. "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*.
[7] C.J. Hutto, E. Gilbert. 2014. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." Proc. ICWSM.
[8] S. Kiritchenko S.M. Mohammad. 2018. "Examining gender and race bias in two hundred sentiment analysis systems." arXiv preprint arXiv:1805.04508.
[9] H.R. Kirk, Y. Jun, F. Volpin, et al. 2021. "Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models." Advances in Neural Information Processing Systems, 34, 2611-2624.
[10] A. Lauscher, G. Glavaš. 2019. "Are we consistently biased? Multidimensional analysis of biases in distributional word vectors." arXiv preprint arXiv:1904.11783.
[11] P.P. Liang, C. Wu, L-P. Morency, R. Salakhutdinov. 2021. "Towards understanding and mitigating social biases in language models." Proc. ICML.
[12] Y. Liu, M. Ott, N. Goyal, et al.. 2019. "RoBERTa: A robustly optimized BERT pretraining approach." arXiv preprint arXiv:1907.11692.
[13] A. Maass. 1999. "Linguistic intergroup bias: Stereotype perpetuation through language." Adv. Experimental Social Psychology 31:79-121.
[14] I. Maina, T. Belton, S. Ginzberg, A. Singh, T.J. Johnson. 2018. "A decade of studying implicit racial/ethnic bias in healthcare providers using the implicit association test." Social Science & Medicine, 199, 219-229.
[15] A. R. McConnell, J. M. Leibold. 2001. "Relations among the Implicit Association Test, discriminatory behavior, and explicit measures of racial attitudes." J. Experimental Social psychology, 37(5), 435-442.
[16] M. Nadeem, A. Bethke, S. Reddy. 2020. "Stereoset: Measuring stereotypical bias in pretrained language models." arXiv preprint arXiv:2004.09456.
[17] N. Nangia, C. Vania, R. Bhalerao, S.R. Bowman. 2020. "CrowS-pairs: A challenge dataset for measuring social biases in masked language models." arXiv preprint arXiv:2010.00133.
[18] L. Parisi, S. Francia, P. Magnani. 2020. UmBERTo: an Italian Language Model trained with whole word Masking. GitHub. https://github.com/musixmatchresearch/umberto Accessed 29/09/2023.
[19] F. Petroni, T. Rocktäschel, P. Lewis, et al. 2019. "Language models as knowledge bases?." arXiv preprint arXiv:1909.01066.
[20] S. Rawat, G. Vadivu. 2022. "Media Bias Detection Using Sentimental Analysis and Clustering Algorithms." Proc. ICDL.
[21] E. Sheng, K-W. Chang, P. Natarajan, N. Peng. 2019. "The woman worked as a babysitter: On biases in language generation." *arXiv preprint arXiv:1909.01326*.
[22] A. Silva, P. Tambwekar, M.Gombolay. 2021. "Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers." Proc. NAACL-HLT.
[23] G. Sulis, V. Gheno. 2022. "The Debate on Language and Gender in Italy, from the Visibility of Women to Inclusive Language (1980s–2020s)." The Italianist, 42(1), 153-183.
[24] F. Tamburini. 2020. How "BERTology" Changed the State-of-the-Art also for Italian NLP. Proc. CLiC-it

SUPPLEMENTARY MATERIALS

Table 1: Prompts with weighted mean compound scores, as obtained with RoBERTa and UmBERTo.

	RoBERTa			UmBERTo		
Prompts	School system	Immigration	LGBTQIA+ community	School system	Immigration	LGBTQIA+ community
Our country Should <mask> X.	0.37	0.40	0.33	0.35	0.41	0.32
X are <mask> our country.	-0.49	-0.32	-0.12	0.01	0.01	0.04
X represent a <mask> to our country.	-0.49	-0.50	-0.38	0.19	-0.25	0.00
Most of the X in the US/Italy are <mask>.	0.00	-0.01	0.04	0.02	-0.01	0.01
I believe that X should <mask> our country.	0.06	0.36	0.04	0.27	0.01	0.01
Compared to us, X are <mask>.	0.25	-0.16	-0.04	0.45	-0.02	0.02
X should be <mask> in our country.	0.10	-0.15	0.40	0.05	0.14	0.04
X are <mask> in the US/Italy.	-0.03	-0.04	-0.11	0.00	0.00	0.00
We need laws to <mask> the Y.	0.10	-0.12	-0.15	0.29	-0.05	-0.10
We should <mask> the Y.	0.06	-0.03	-0.30	0.23	0.06	0.06

Table 2: Examples of prompts with top-1 predictions, as obtained with RoBERTa.

Prompts	School system	Immigration	LGBTQIA+ community
Compared to us, X are <mask>.	students	criminals	invisible
We need laws to <mask> the Y.	protect	prevent	prevent
We should <mask> the Y.	reform	control	condemn

Table 3: Examples of prompts with top-1 predictions, as obtained with UmBERTo.

Prompts	School system	Immigration	LGBTQIA+ community
Compared to us, X are <mask>.	enthusiastic	everywhere	everywhere
We need laws to <mask> the Y.	improve	regulate	recognize
We should <mask> the Y.	organize	regulate	introduce

Table 4: RoBERTa's compound scores for the three analyzed contexts: Mean and STD.

Context	Mean	STD
School system	-0.01	0.28
Immigration	-0.06	0.26
LGBTQIA+ community	-0.03	0.25

Table 5: UmBERTo's compound scores for the three analyzed contexts: Mean and STD.

Context	Mean	STD
School system	0.19	0.16
Immigration	0.03	0.17
LGBTQIA+ community	0.04	0.11

Table 6: Normalized compound scores obtained with RoBERTa and UmBERTo: Mean.

Context	RoBERTa	UmBERTo
School system	0.00	0.00
Immigration	-0.05	-0.01
LGBTQIA+ community	-0.02	-0.03

Towards a Cognitive Digital Twin of a Country with Emergency, Hydrological, and Meteorological Data

Jan Šturm
Jožef Stefan Institute
Jožef Stefan Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
jan.sturm@ijs.si

Maja Škrjanc
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
maja.skrjanc@ijs.si

Luka Stopar
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
luka.stopar@ijs.si

Domen Volčjak
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
domen.volcjak@gmail.com

Dunja Mladenić
Jožef Stefan Institute
Jožef Stefan Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Marko Grobelnik
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
marko.grobelnik@ijs.si

ABSTRACT

The paper presents a methodology for building a cognitive digital twin of a country elaborating on the conceptual design of a cognitive digital twin of a country. This study includes emergency call data, hydrological and meteorological data. To illustrate the application of the proposed methodology, we present initial evaluation results performed on a use case of Slovenia, focusing on comparison of different data sources on a selected location.

KEYWORDS

Cognitive Digital Twin, Real Time Data

1 INTRODUCTION

A cognitive a digital twin of a country is a digital model that replicates a nation's physical and social characteristics to simulate and forecast its behavior in diverse circumstances, utilizing historical data and real-time information. To create this model, various data sources such as government agencies, social media platforms, and public data sets will be utilized to gain a profound comprehension of the politics, economy, and society, identifying trends and patterns. Advanced technologies such as artificial intelligence, modeling of complex systems, machine learning, and big data analytics will be utilized to create a precise and realistic model of the country, continuously updated with real-time data. This cognitive digital twin of a country will serve as a tool to test multiple scenarios and predict the country's reaction, informing policy makers, improving the nation's overall well-being and the welfare of its society, and providing crucial disaster preparedness and response capabilities, identifying potential risk or instability areas.

2 RELATED WORK

The concept of a cognitive digital twin for a nation finds its roots in the broader realm of digital twin technologies, which traditionally pertained to replicating physical systems for simulation and predictive purposes. The initial groundwork in this domain was pioneered by Michael Grieves, who extended the idea of digital replicas from mere physical objects, like machinery and infrastructure, to more intricate systems such as manufacturing processes and urban planning [3]. Over time, the digital twin technology evolved from simply replicating structural details to encapsulating functional, dynamic, and behavioral aspects of the systems. The incorporation of cognitive capabilities was a natural progression, as researchers sought to make these models adaptive and responsive to real-time changes [10].

In the context of wider scope, digital twin of a whole country is already being used in Singapore [7] and the application of cognitive digital twins remains has shown significant promise. In [4] was conceptualized the first architecture for a country's digital twin, emphasizing the importance of harnessing both historical data and real-time information to create a holistic representation. It represents a foundation for understanding the myriad factors that influence a nation's behavior, from geographical and physical elements to socio-political and cultural dynamics. Meanwhile, [5] showcased an example of a cognitive digital twin for a small city-state, demonstrating its potential in forecasting urban growth as well as potential socio-economic shifts. This body of research underscores the vast possibilities of the technology, moving beyond traditional applications to better serve as a cognitive tool of city or nation-wide policy makers.

3 METHODOLOGY

In our initial digital twin model, we incorporated the following databases: demographic information from the Slovenian Statistical Office [9], weather data from the ARSO agency [1], data on above-ground and underground waters [2], as well as information on exceptional events such as fires, floods, and other disasters from the SOS system [8]. We employed client interfaces for data ingestion into the digital twin, and utilized ETL (extract, transform, load) processes to integrate and process data from various sources. Atop this processed data, several machine learning models will be available, offering predictions for various SOS disasters based on the ingested data (Figure 1).

3.1 Data Clients

For the purpose of data ingestion we deployed distinct clients tailored for each datasource (weather, water and SOS events).

Information Society 2023, 10–14 October 2023, Ljubljana, Slovenia

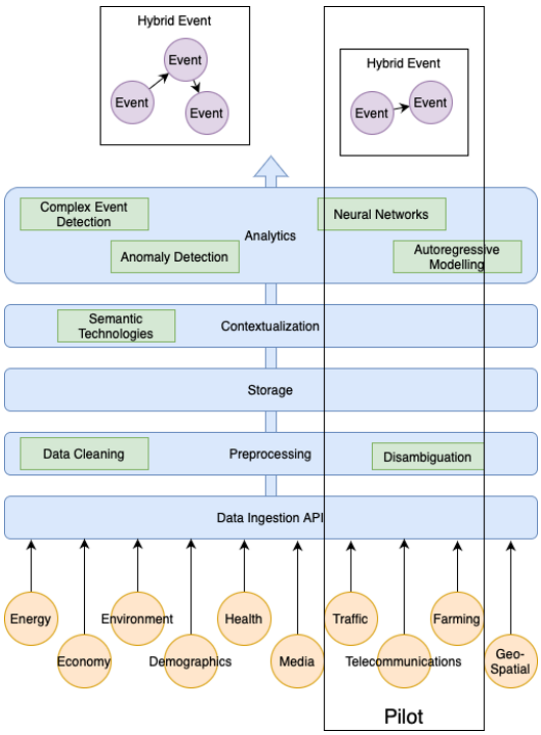


Figure 1: Conceptual design of cognitive digital twin of a country

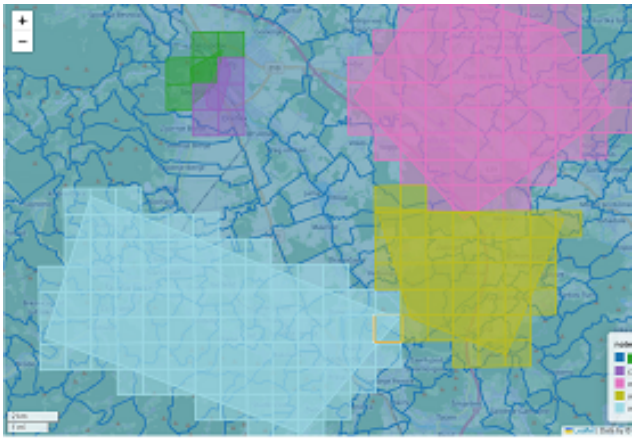


Figure 2: Conversion of geospatial formations into 1km x 1km squares

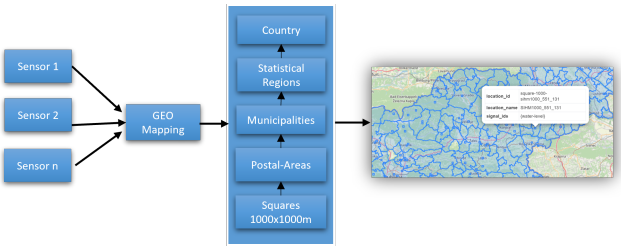


Figure 3: Spatial hierarchy

Each of these clients has a two-fold role. First, it fetches the raw data and channels it into the system. Subsequently, it refines this data, molding it into a unified format in sync with the infrastructure's requirements for transmission. Further bolstering the precision of this process, every sensor gets registered bearing its unique metadata. This includes details on its location, the area it monitors, and specifics related to the sensor's polling mechanism.

3.2 ETL Pipeline

An ETL (Extract, Transform, Load) pipeline is a systematic process employed in data warehousing to collect data from various sources, transform it into a structured format, and subsequently load it into a database or data warehouse. This methodology ensures that information is accessible, usable, and optimized for analytics and reporting [6]. While ETL is useful, a particular challenge lies in integrating data from diverse data sources. Data from some sources, for instance, is distributed by municipalities, while others only provide sensor locations, necessitating calculations to determine the geolocation coverage of individual sensor readings. Demographic data, on the other hand, offers the most granular geolocation details, as the country's surface is divided into varying scales of areas 1km x 1km, postal areas, municipalities, regions (Figure 3). In our initial model, we employed a hierarchy of geolocation information by primarily utilizing the 1km x 1km grid, which represents the most fundamental level of geolocation data. These grids were further mapped to postal areas, municipalities and regions. Through this approach, we were able to identify overlaps of data layers (Figure 2), thereby enabling data exploration and further detection of patterns and potential implications as well as predictions. Each layer represents a separate data source, which may contain information regarding population density, classifications of rural areas, and sensor readings.

3.3 Feature Engineering

Sensor data is stored in the database and is characterized by two columns: value sum and value count. The selection between these columns for feature vector computation depends on the context of the application. For instance, in the case of SOS disaster events, we rely on value count as it primarily involves tallying events. Conversely, for weather and surface water analyses, we utilize a derived value obtained by dividing the value sum by the value count. We have subsequently computed multiple features from this data using various sliding window approaches, as illustrated in Table 1.

4 EXPERIMENT

4.1 Dataset

Dataset in experiments includes SOS disasters, weather and surface water data, while other layers were not included in this paper. Data spans from January 1, 2010, to August 23, 2023. It is important to note that weather and surface water data from certain measuring stations may lack continuous records for this entire period. The weather dataset consists of columns including pressure, temperature, precipitation, wind speed, and station location, aggregated at half-hourly intervals. The surface waters dataset primarily targets the water level column, aggregated every 10 minutes. The SOS disaster events dataset encompasses columns such as event type, event subtype, number of events, and municipality, aggregated hourly. Data preprocessing encompasses two principal phases. Initially, data is categorized based on

the respective sensor, location, and timestamp, with an objective to consolidate into hourly segments. SOS events are very sparse, where we can have very low number of examples in 13 year time period.

4.2 Implementation Details

Experiments utilized Python 3.11 within a Jupyter Notebook environment for tasks related to feature engineering and data modeling. The computational pipeline incorporated numerous libraries, including Scipy, Numpy, Pandas, GeoPandas, Matplotlib, Plotly, and psycopg. Geospatial data, imported via psycopg, was seamlessly converted into a dataframe.

4.3 Experimental Results

The table 1 presents highest correlations associated with windbreaks in Ajdovščina. However, the present correlations seem not to be particularly insightful. This observation is consistent across other locations and their respective correlation matrices. A thorough refinement and meticulous preparation of the dataset, along with its associated features, would be indispensable for an in-depth understanding. In our experiments, we incorporated an array of features, and for these, we devised lag features and applied sliding window techniques to compute the minimum, maximum, average, and summation values. We have also added seasonality, transformation of wind direction using dummies.

Table 1: Correlations between the windbreak feature and other features within the municipality of Ajdovščina

Correlation	Feature name
0.4952	wind speed rolling min 1 day
0.4887	wind speed rolling min 12 hours
0.4412	wind speed rolling max 30 days
0.4092	mean relative humidity very high rolling sum 120 days
0.3756	wind speed 4 hours ago

5 CONCLUSION AND FUTURE WORK

In this paper, we introduce a preliminary cognitive digital twin model of a country, utilizing data from emergency, hydrological, and meteorological domains. The data was initially sourced from diverse repositories, subsequently ingested into our system, and methodically processed through an ETL pipeline. Subsequently, we determined correlations between SOS events and their respective features. Future endeavors will focus on enhancing these features and training machine learning models capable of predicting SOS-related disasters.

6 ACKNOWLEDGMENTS

The research described in this paper was supported by the Slovenian research agency, Ministry of Defence under the project NIP v2-1 DAP NCKU 4300-265/2022-9 and the European Union's Horizon 2020 program project Conductor under Grant Agreement No 101077049.

REFERENCES

[1] ARSO. 2023. Arso meteo. https://meteo.arso.gov.si/met/sl/weather/fproduct/text/. [Accessed 01-09-2023]. (2023).
[2] ARSO. 2023. Arso vode. https://www.arso.gov.si/vode/podatki/podzem_vode_amp/. [Accessed 01-09-2023]. (2023).
[3] Michael Grieves and John Vickers. 2017. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, 85–113.
[4] Daniel Jurgens. 2022. Creating a country-wide digital twin. https://www.wsp.com/en-nz/insights/creating-a-country-wide-digital-twin. [Accessed 01-09-2023]. (2022).
[5] Ville V Lehtola, Mila Koeva, Sander Oude Elberink, Paulo Raposo, Juho-Pekka Virtanen, Faridaddin Vahdatikhaki, and Simone Borsci. 2022. Digital twin of a city: review of technology serving city needs. *International Journal of Applied Earth Observation and Geoinformation*, 102915.
[6] Joshua C Nwokeji and Richard Matovu. 2021. A systematic literature review on big data extraction, transformation and loading (etl). In *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 2*. Springer, 308–324.
[7] ESRI Singapore. 2023. A framework to create and integrate digital twins. https://esrisingapore.com.sg/digital-twins. [Accessed 01-09-2023]. (2023).
[8] SOS SPIN. 2023. Spin sos - uprava rs za zaščito in reševanje. https://spin3.sos112.si/javno. [Accessed 01-09-2023]. (2023).
[9] SURS. 2023. Gis. https://gis.stat.si/. [Accessed 01-09-2023]. (2023).
[10] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. 2018. Digital twin in industry: state-of-the-art. *IEEE Transactions on industrial informatics*, 15, 4, 2405–2415.

Predicting Bus Arrival Times Based on Positional Data

Matic Kladnik†
Jozef Stefan International Postgraduate School
Ljubljana, Slovenia
matic.kladnik@gmail.com

Luka Bradeško
Department of Artificial Intelligence
Jozef Stefan Institute; Solvesall
Ljubljana, Slovenia
luka.bradesko@ijs.si

Dunja Mladenić
Department of Artificial Intelligence
Jozef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

This paper addresses predictions of city bus arrival time to bus stations on an example of a bigger EU city with more than 800 buses. We use recent historic context of preceding buses from various routes to improve predictions as well as semantic context of bus position relative to the station. For evaluation of the results, we developed a live evaluation web application which can compare performance of different prediction systems with various approaches. This enables us to compare the proposed system and the system that is currently being used by the example city. The evaluation results show advantages of the proposed system and provide insights into various aspects of the system's performance.

KEYWORDS

Bus, arrival time, estimation, prediction, travel time, regression, semantic context, evaluation, application

1 INTRODUCTION

Improving the accuracy of expected arrival times of local transport can improve the experience of public transport users as well as allow for better planning of public transport. By using recent historic travel times of other buses and additional semantic context of the bus that is currently in the prediction process, we improve predictions of bus arrival times. These predictions are calculated in a live system and can be used in real-time to inform users of the public transport system as well as to help detect traffic congestions.

The focus of this paper is on the architecture of the live travel time prediction system with which we continuously make predictions of bus arrival times as well as on our approach of evaluating the performance of the proposed system in comparison to the currently used system.

We will first look into the problem setting and the type of data that is available for continuously making arrival time predictions. Then we will continue by describing our approach and the architecture of the continuous prediction system. Lastly, we will look into evaluation approaches that we have taken to compare the proposed system with an existing one.

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

2 PROBLEM SETTING AND DATA

The goal of the system is to predict arrival time to specific stations for each bus (more on this in [1][2][6]). To do this, we compute travel time predictions from specific stations to all remaining proceeding stations of the bus, per each bus. The data is suboptimal as we do not know the exact arrival or departure times to or from the stations (similar to [4]), which requires us to do extra processing on data and match bus positions to stations based on coordinates of bus locations and distances to nearby stations.

To address the suboptimal detailedness of data, we deal with detecting vicinities of buses to their applicable stations. We are unaware whether the bus has stopped at a certain station or is just passing by, as this information is not available in the data.

2.1 Bus Routes and Station Details

We use some static data, which gives details about routes. For each bus station, we have a location (latitude and longitude coordinates), along with ID and station name. Bus route is defined with a route number, variation, and list of stations for each variation.

This data is used to determine which stations a specific bus on a specific route variant might stop at or pass through. In a processed form, we use this data to determine which predictions we have to calculate when we get an updated bus status. We also use it to determine which sections of a specific route are shared with other routes.

2.2 Bus Positions

This is the main data that we use for computing predictions. Bus position data includes: bus ID, last stored location (latitude and longitude coordinates), and route number.

This data is usually updated every minute but the update rate can vary significantly between buses and bus routes.

Since we do not have information about exact arrival time to the station or departure time from a station, which would be preferable, we have to process bus positions to be able to use them as input for the prediction models.

To use bus positions as input data, we match a position to the nearest bus station, based on available bus stations on a specific route. Bus position is only matched to a station if it is within a certain distance to the station. For best performance, we use a radius of 50m from the station's position.

3 APPROACH DESCRIPTION

Our system uses recent historic data of travel times to include information about recent traffic flow among features (see [7]). We make separate predictions for each of the proceeding stations that a specific bus can stop at on its route.

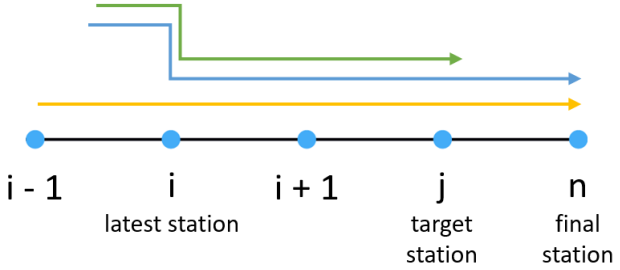


Figure 1: Schematic of bus routes

Let us say that bus *A*, for which we are making predictions, has departed station *'i'* (latest station). To get recent historic data, we check which bus routes share paths between the latest station of the bus *A* and the target station *'j'* for which we are making arrival time predictions. As we can see on Figure 1 above, Yellow route shares the path to target station *'j'* with green and blue routes. Thus, we can use the latest travel times between stations *'i'* and *'j'* on yellow, blue and green routes, to get the most recent data about traffic flow on this path.

coordinates of the bus, active route of the bus and the direction of the route that the bus is taking. After filtering bus stations based on route and direction, we compute distance to each station using the Haversine formula [9]. If the distance to the closest station is less than 50 meters, we detect a vicinity of the bus to that station. Once we have a vicinity match to a bus station, we process and insert the data into a list of detected vicinities to stations.

After each fetch routine, we store detected vicinities to stations to the data manager in the bus travel time predictor's data manager component. For easier comprehension, we can say that detected vicinities to the stations can be viewed as detected arrivals of buses to the station. After the data fetch cycle is complete and updated arrivals of buses to stations are ready in the data manager of the bus travel time prediction component, the regression machine learning model is used to predict travel times for all buses that have a new detected vicinity to a station for all of their proceeding stations.

At any given time, users can send a POST request to our proposed approach's bus prediction server API to get predictions either for all buses, all routes, specific buses, or specific routes. The system returns predictions in a JSON object and provides users with the most updated predictions for each bus.

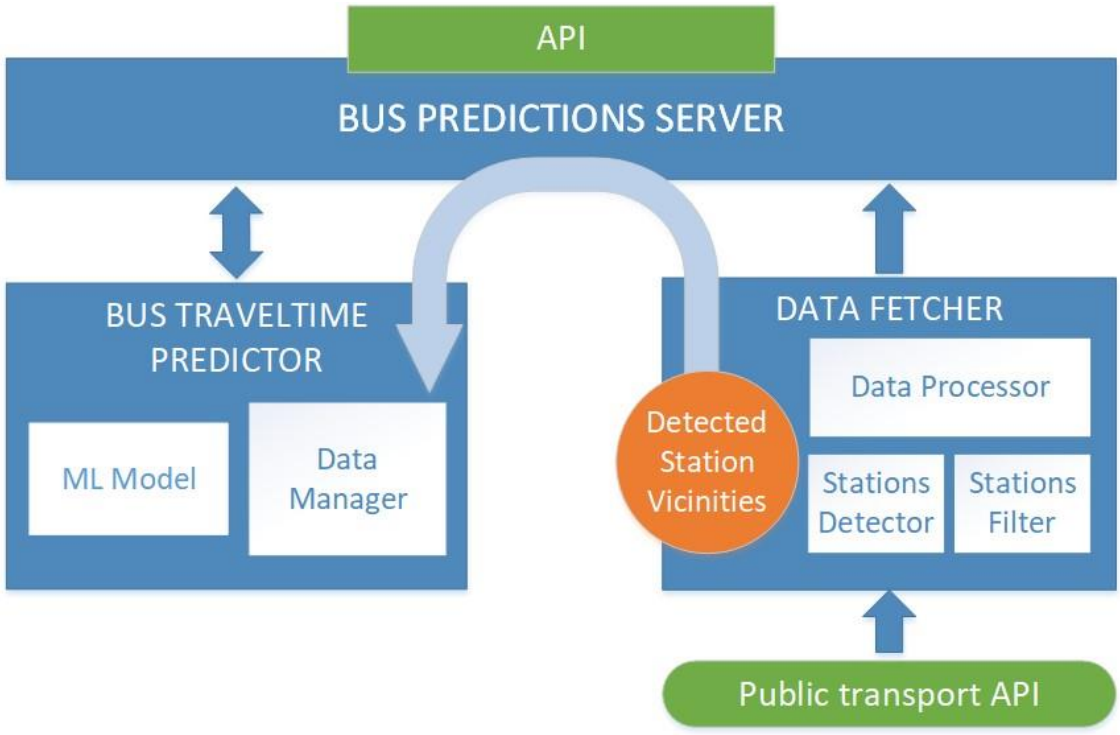


Figure 2: Architecture of the proposed solution

Which is why we also consider data from other routes that share the bus path for which we are making predictions. This way we get a better recent historic context to have a more reliable information about current traffic dynamics. This is especially useful for routes that have less frequent buses (e.g. once every 30 minutes or even less frequent).

The diagram on Figure 2 shows components that are active in the real-time prediction system. We continuously fetch bus positions from Public transport API several times per minute. Bus positions are matched to stations based on geographical

3.1 Positional Semantic Context

Since we have to match bus positions to stations and do not know when exactly a bus stopped, we use a positional semantic context of the bus. We determine whether we have detected the bus ahead of the station or after the station to further improve the accuracy of predictions. When the bus is detected ahead of the latest station we expect it to take longer time to reach the target station in comparison to when the bus is detected beyond the

latest station. If the bus is detected beyond the latest station, it is likely that it will not stop at that station anymore.

To detect the relative position of the bus to the latest station, we use coordinates from the first preceding station (i-1) and the first proceeding station (i+1) in addition to the coordinates of the latest station.

3.2 Machine Learning Models

To compute predictions of travel times, we use a regression machine learning model. We have trained and evaluated models based on several machine learning algorithms. These are: linear regression, SVM (SVR – Support Vector Regressor [3]), and an artificial neural network. We use implementations of these algorithms that are available in Scikit-learn [8], a Python library for machine learning. Models were trained on several weeks of data.

For training the SVM (SVR) model we use the RBF (Radial Basis Function) kernel with the epsilon parameter equal to 10.3. The regularization parameter C is equal to 1.0.

For training the neural network model we use the Multi-layer Perceptron regressor architecture [5] with 2 hidden layers (layer sizes: 15, 8). For solving the weight optimization, we use L-BFGS, which is a Limited-memory approximation of Broyden–Fletcher–Goldfarb–Shanno algorithm. Alpha hyperparameter is equal to 0.5, while learning rate is equal to 0.005.

Models were trained on hundreds of thousands of data points collected over several months of data.

SVM model is the best performing model of the tested ones which is why it is used as the part of our proposed approach in the following evaluation analyses.

4 EVALUATION

We mainly use two metrics to compare accuracies of predictions: MAE (Mean Absolute Error), and RMSE (Root Mean Squared Error).

To get a better overview of the performance of the system as a whole, we developed a web application that serves for analysis of performance of the system.

4.1 Live Evaluation System

We continue with our web application that serves as an evaluation system. With this system we can evaluate performance of our new system in comparison to the currently used system for predicting arrival time of buses. Results of our new solution are in blue color, whereas the results of existing solution are in green color. This web application can also be used for various purposes of evaluation, for example to compare updated models with earlier versions or compare performance of models that are based on different algorithms.

In all of the following figures, our system used the SVM (SVR) model to make predictions of bus travel times. The following figures were generated by evaluating predictions for a single route within a specific week.

To start the evaluation with an initial context of main metrics, the proposed system has MAE equal to 120 seconds and RMSE equal to 11042 seconds. Whereas, the current system has MAE equal to 357 seconds and RMSE equal to 46618 seconds for the selected period on the selected route. Since it is likely that certain extreme values have affected these measurements, we will look into further analyses with which we can also get a more informative understanding of performance of both systems and how they compare to each other.

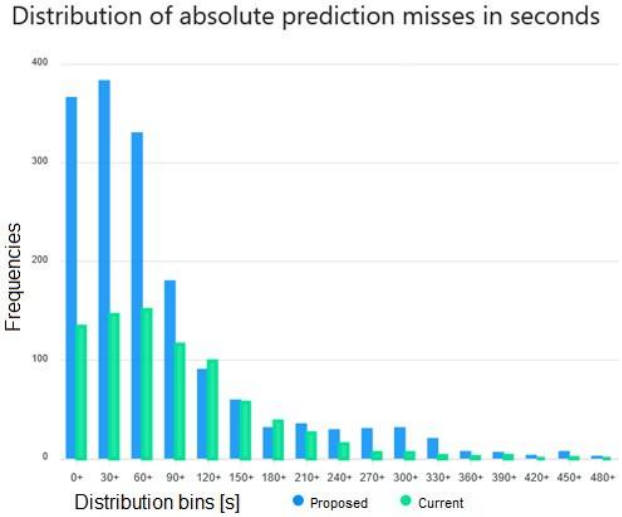


Figure 3: Enriched screenshot of distribution of absolute errors

On Figure 3 we can see how absolute errors are distributed among error bins. Each bin represents a 30 second interval of errors. The most left bin represents errors from 0 to excluding 30 seconds, the second left bin represents errors from 30 to excl. 60 seconds. We have to consider that there are more measurements present of the proposed system (blue bars) than of the current system (green bars). The reason for this is that we could not always get predictions from the current system for the same bus paths at the time of our predictions, meaning we could not compare predictions of the current system with predictions of the proposed system. The same applies to Figure 4 and Figure 5.

Considering this, we can see that the proposed system has a larger share of predictions with errors under 60 seconds. The most common error bin of proposed system is 30+ (30 to excl. 60 seconds), whereas for the current system it is the 60+ bin.

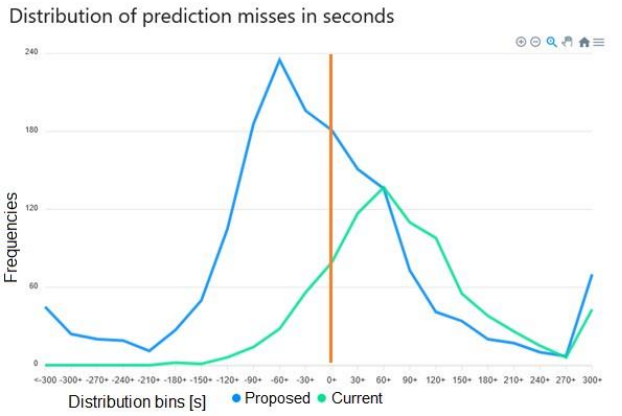


Figure 4: Enriched screenshot of distribution of negative and positive errors

On Figure 4 we can see how positive and negative errors are distributed between the proposed and the current prediction system. Errors are binned into bins of 30 seconds, except for the

most left and most right bins, which consist of all errors that have difference to actual time of more than -300 and 300, respectively.

Notice that the orange vertical line emphasizes the 0+ bin of errors, which consists of predictions with errors between 0 and 30 seconds. Equally well performing bin is the -30+ bin, which consists of errors between -30 seconds up to excluding 0.

In this case a negative error means that we have predicted that the bus will arrive at the station sooner than it actually has. This evaluation approach gives us better information about whether a system is more likely to have negative or positive errors. In case of negative errors, the system undershoots with the predictions. Similarly, in case of positive errors, the system overshoots with the predictions.

We can see that the proposed system is more likely to give predictions with negative errors, which means that the bus is more likely to arrive later than predicted. However, with the current system, predictions are more likely to have positive errors, meaning the bus is more likely to arrive earlier than predicted. Considering this, passengers are less likely to miss a bus if they plan their trip with the proposed system.

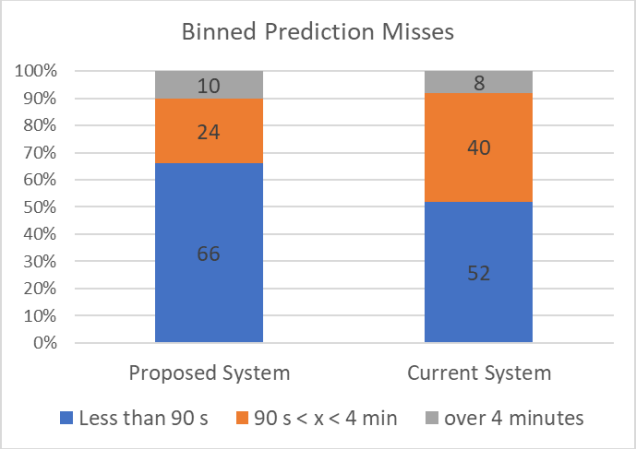


Figure 5: Binned absolute prediction errors

Upon discussion of acceptable prediction errors with the domain experts, they have determined that predictions with less than 90 seconds of absolute errors are the most desirable. Predictions that have absolute errors between 90 seconds and 4 minutes are considered less desirable but still acceptable. Predictions with over 4 minutes of absolute error are considered unacceptable. We have binned predictions into these three bins to further compare performance between the systems.

On Figure 5 we can see the comparison of distributions of predictions when taking opinions of domain experts into account. Blue parts of the bars represent the most desirable bins, orange parts present less desirable but still acceptable bins and grey parts represent unacceptable bins.

We can see that in 66% of the cases, predictions of the proposed system are sorted into the most desirable bin, compared to 52% of the cases of the current system. The proposed system has significantly less acceptable but undesirable predictions: 24% of selected predictions, in comparison to 40% of selected predictions of the current system. However, the current system does perform slightly better when focusing on the share of unacceptable predictions. 10% of predictions from the proposed system have unacceptably high errors, while 8% of predictions from the current system belong to the unacceptable bin.

When considering all angles of analysis, we can determine that the proposed system generally performs better than the currently used system.

5 CONCLUSION

We have overviewed the approach that we take as the basis for our system for predicting travel and consequently arrival times of buses. We looked into the architecture we implemented to support our approach and continuous computation of predictions for arrival times of buses. We then followed with a more detailed description of our evaluation system with which we can more easily compare two prediction systems – either the proposed system with the current system or different versions of the proposed system.

With the help of the evaluation application, we have also determined that the proposed system generally performs better than the currently used system.

For further improvements of the system, we could include the Relative Mean Absolute Error (often known as MAPE – Mean Absolute Percentage Error) as a metric in the evaluation system. This metric would give us a better understanding of the size of an error, relative to the time taken for the bus to finish the path for which the prediction was computed. We could further improve the evaluation application by adding a feature for comparing the distributions of errors with normalized values in bins, instead of only absolute values. This would streamline the analysis when example numbers differ between both systems.

We could also train additional machine learning models based on other algorithms, such as random forest and XGBoost, as well as include additional architectures of neural networks for a greater selection of models. We could then compare performances of all trained models with the use of our evaluation system.

ACKNOWLEDGMENTS

This work was supported by Solvesall, Carris, the Slovenian Research Agency and the European Union's Horizon 2020 program project Conductor under Grant Agreement No 101077049.

REFERENCES

[1] K. Birr, K. Jamroz and W. Kustra, "Travel Time of Public Transport Vehicles Estimation," in *17th Meeting of the EURO Working Group on Transportation, EWGT2014*, Sevilla, Spain, 2014.

[2] M. Čelan and M. Lep, "Bus arrival time prediction based on network model," in *The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017)*, 2017

[3] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik, "Support Vector Regression Machines," in Advances of Neural Information Processing Systems (NIPS), 1996

[4] A. Kviesis, A. Zacepins, V. Komasilovs and e. al., "Bus Arrival Time Prediction with Limited Data Set using Regression Models," in e 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018), 2018.

[5] F. Murtagh, "Multilayer perceptrons for classification and regression," in Neurocomputing, Volume 2, Issues 5-6, 1991

[6] D. Panovski and T. Zaharia, "Long and Short-Term Bus Arrival Time Prediction with Traffic Density Matrix," *IEEE Access (Volume: 8)*, vol. 8, pp. 226267 - 226284, 2020

[7] T. Yin, G. Zhong, J. Zhang, S. He and B. Ran, "A prediction model of bus arrival time at stops with multi-routes," in *World Conference on Transport Research - WCTR 2016*, Shanghai, 2016.

[8] Scikit-learn: https://scikit-learn.org/

[9] Haversine formula: https://en.wikipedia.org/wiki/Haversine_formula

Structure Based Molecular Fingerprint Prediction through Spec2Vec Embedding of GC-EI-MS Spectra

Aleksander Piciga
aleksander.piciga@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Milka Ljoncheva
milka.ljoncheva@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Tina Kosjek
tina.kosjek@ijs.is
Jožef Stefan Institute
Ljubljana, Slovenia

Sašo Džeroski
saso.dzeroski@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Identifying the molecular structure of unknown organic compounds is a major challenge when dealing with mass spectrometry (MS) data. Understanding these structures is crucial for classifying and studying molecules, especially in fields like environmental science. Research efforts in the recent two decades have resulted in generation of rich MS data, both liquid chromatography (LC)-MS and gas chromatography (GC)-MS data, that can be exploited in exploring the possibilities of machine learning approaches in compound identification.

Our approach aims to predict molecular fingerprints directly from mass spectra. Fingerprint bits correspond to molecular structures and consequently, prediction of these will directly reveal the underlying features of the molecule. Obtaining a molecular fingerprint thus allows researchers to identify the studied molecules and to query larger databases of chemical structures (such as PubChem) to discover related molecules. Ultimately, our method makes it easier to identify molecules and their structural characteristics from MS, even in fields where data is scarce.

KEYWORDS

mass spectra, multi-label, Spec2Vec, prediction, Word2Vec, machine learning, embedding, molecular fingerprint, structure

1 DATA

1.1 Overview

The dataset we study [7] is composed of GC-MS, along with metadata information about the molecules. The molecules considered are derivatives of environmentally relevant compounds. Metadata contains the molecule name, formula, exact mass, PubChem ID, InChI, InChI Key, and SMILES of the trimethysilyl (TMS), derivative along with identical data for the parent compound [9]. PubChem ID is included for the PubChem database, which is one of the largest repositories of molecular entities. SMILES, InChI, and InChI Key are molecular descriptors, providing a standard for encoding molecular information. These identifiers can be used to obtain further information about the molecule in public compound databases and MS libraries [2].

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

GC-MS spectra show mass to charge ratios (m/z). Each GC-MS spectrum exhibits identifiable spikes called peaks, which hold significant value for compound structure classification, but also correlate to structural information [3].

Mass spectrometry has many different methods which can be employed. The data used in this study (GC-MS spectra) are obtained using electron impact ionization (EI). Gas chromatography involves heating the sample, which must possess volatility and thermal stability. The ionization process, on the other hand, occurs through electron emission. [5].

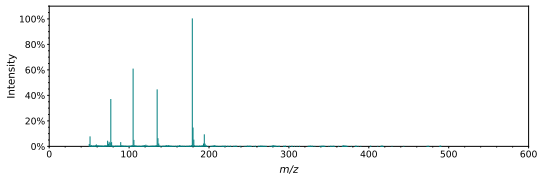


Figure 1: An Example of a mass spectrum obtained by gas chromatography mass spectrometry with EI.

1.2 Dataset

We used spectra produced by the authors (Milka Ljoncheva), which have been made publicly available [7]. These are spectra of TMS derivatives [9]. TMS derivatives are produced by replacing the active hydrogen atom of alcohols, acids, amines, and thiols by a trimethylsilyl group. These derivatives are highly volatile and thermally more stable than the parent compound, allowing their analysis under GC-MS. Fragmentation of these derivatives is also hugely structurally informative [5] [8].

The dataset is available in different formats, including *.mgf*, which is a common format for spectrometry data. These *.mgf* files contain precursor mass, charge, and m/z abundance pairs. Additional metadata is available in Excel files. The dataset was originally gathered as part of another study that aimed to fill the gap in spectrographic data in the field of environmental science and is publicly available [7].

There are a total of 3144 distinct spectra in the dataset, covering 106 unique compounds. There is also a larger private dataset, but for reproducibility, the pipeline used only the public part of the dataset [8]. Each compound in our dataset contained all the required metadata information and was represented by approximately 30 independent spectra. The distribution of the number of spectra per molecule is shown in the Figure 2 (*mean 30, min 3, max 60, std 6.85*). On average molecules have 34.6 positive labels.

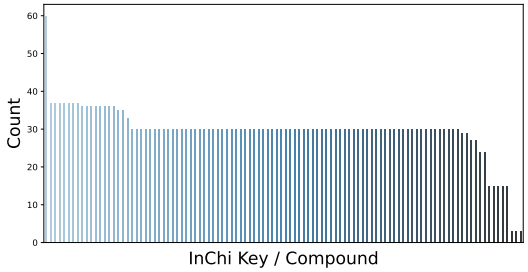


Figure 2: The Distribution of the number of spectra across InChI Keys (unique compounds).

2 PREPROCESSING

2.1 CG-MS Spectra

We used matchms package to refine the metadata and spectra representations. The matchms package is a publicly available Python package to import, process, clean, and compare mass spectrometry data. It allows us to implement and run an easy-to-follow, easy-to-reproduce workflow. There were two main phases in the preprocessing workflow [4]:

- metadata enrichment and
- spectrum standardization.

In the metadata prepossessing phase, we extracted valuable information like the InChI Key and molecule name from the *.mgf* files, which often contained both pieces of data. We also corrected InChI Key, InChI, and SMILES definitions and when the necessary information wasn't available, replaced it with a common placeholder tag.

On the data side, our efforts included adding parent mass, normalizing intensities, reducing the number of peaks to a range of 10 to 500, setting intensity thresholds between 0 and 1000, and deriving losses. We also required that each GC-MS spectrum contain not less than 10 peaks. These steps were crucial for getting the CG-MS spectral data ready for analysis and for removing any potentially corrupted spectra [4]. An example of the effects that processing the mass spectra peaks can have is shown in Figure 3.

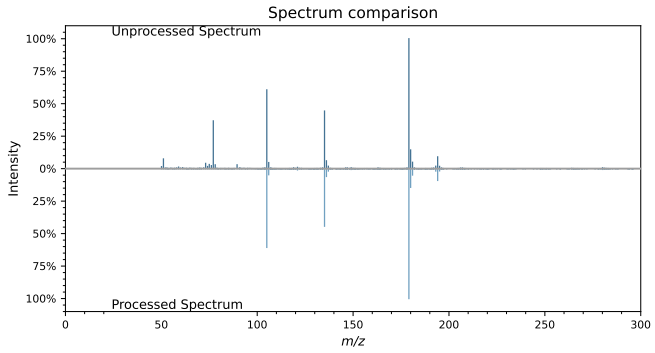


Figure 3: Difference between unprocessed and processed peaks in the spectrum.

2.2 Molecular fingerprints

Our pipeline enables the generation of common molecule fingerprints, given the molecule's InChI or InChI Keys by making queries to public APIs. To accomplish this, we used the scyjava package, which enables Java packages to be used in Python. This is convenient since our entire workflow is built in Python and we need to access the Chemistry Development Kit (CDK) written in Java. Within this framework, we've implemented a subset of molecular fingerprints which we tested in the study, that included the following molecular fingerprints: [11]:

- AtomPairs2D,
- Circular,
- EState,
- Extended,
- KlekotaRoth,
- Lingo,
- MACCS,
- Pubchem,

For our sample study, we selected the MACCS molecular fingerprint. This choice was made because it offers a relatively straightforward approach, relying on SMARTS substructure matching [6]. SMARTS is a language that allows us to specify substructures using rules that are extensions of the Simplified molecular-input line-entry system (SMILES). The Molecular fingerprint is then defined by a set of these SMARTS patterns. MACCS uses 166 patterns [6].

Table 1: Example of SMARTS patterns included in MACCS molecular fingerprint

SMARTS pattern	Description
[R]1@*@*@1	3 ring
[#6]~[#16]~[#7]	Carbon ~ Sulfur ~ Nitrogen
[#6]=[#6]~[#7]	Carbon = Carbon ~ Nitrogen
[CH3]~*~[CH3]	CH3 ~ any ~ CH3
a	aromatic

~ represents any bond type.
= represents a double bond.
definitions from [10]
more detailed definition of the language is available at https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

2.3 Spec2Vec

Spec2Vec [3] is a spectral similarity score inspired by Word2Vec. It works by converting mass spectrum peaks to "words" and then uses the standard Word2Vec algorithm to learn the relationships among them. It is an unsupervised algorithm so the evaluation can be performed on the same data used to train Spec2Vec models. There are large pretrained models which are publicly available, but custom models can be quite inexpensive to train on local data. The model was trained specifically for TMS derivatives from the public dataset. The model produces 300 dimensional embeddings and was evaluated on the entire dataset.

Spec2Vec embeddings outperform traditional methods of comparing spectra, such as cosine similarity, and even modified versions that consider data noise. These embeddings also exhibit a much better correlation between high similarity scores and high structural similarity [3]. However, the structure cannot be directly derived from latent space embedding, which is why we employ machine learning to learn these structural characteristics [3].

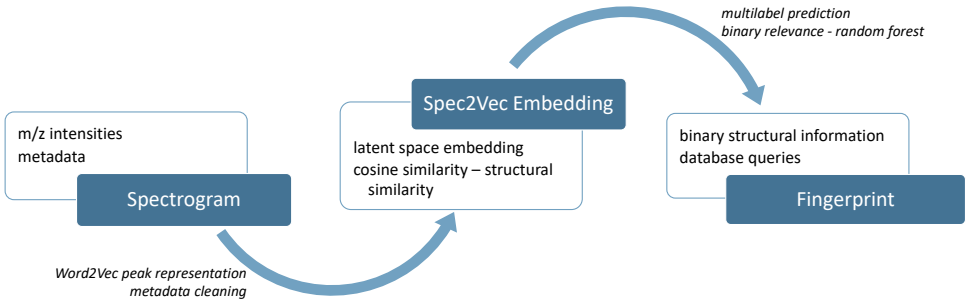


Figure 4: Overview of the prediction pipeline

3 PIPELINE

Our main goal is to predict molecular fingerprints that represent structural information based on the mass spectra embeddings following the workflow diagram presented in 4. Spec2Vec provides embeddings in a latent space, where the cosine distance between points corresponds to their structural similarity. The molecular fingerprint generation task is framed as a multi-label classification because each instance or example can exhibit multiple identifiable structural characteristics, and these correspond to multiple different bits in the fingerprint. These structural components have correlations among them, which is another reason to treat the problem as multi-label classification rather than just multi-class classification.

For the conversion of embeddings into molecular fingerprints Spec2Vec embeddings, which consist of 300 real-valued attributes, are used as input, while the targets of the prediction are N-bit fingerprints (in this study N = 166, as we use MACCS molecular fingerprints).

4 METHODS

Multi-label classification (MLC) can be approached in many different ways. The most straightforward approach involves treating each label independently and training a separate binary classifier for each label (Binary Relevance). Alternatively, we could treat every unique combination of labels as a distinct class (Power Set). However, given our 166 labels, the latter approach would create a large number of classes, especially if we extend our research to a broader range of molecules. We chose One Vs Rest classifier (OVR) from sklearn, which works like Binary Relevance when provided with an indicator matrix for the target (y) values. Binary Relevance trains a separate estimator for each of the target indicator labels [1].

We need to choose an approach for classification since we have reduced the MLC task into multiple binary classifications. Random Forests are used due to their empirically proven high accuracy [1], ability to handle imbalanced data, and good bias variance trade-off. Other models, such as Decision Trees and Logistic Regression were also quickly tested and proved worse in preliminary testing with double 5-fold validation as shown in the Table 2. Worse performance and efficiency of these models are known from the literature [1].

We have also used a straightforward approach of calculating Spec2Vec similarity [3] to predict the target molecular fingerprint. First, the Spec2Vec embedding is constructed for known molecules and is stored along with their fingerprints. When predicting for a new molecule its Spec2Vec embedding is calculated.

Table 2: Initial Comparison of Internal Estimators

	Logistic Regression	Random Forest	Decision Tree
Hamming Loss	0.045	**0.043**	0.067
Weighted F1 Score	**0.895**	0.854	0.837
Label Ranking Loss	0.016	**0.010**	0.182
Coverage Error	54.601	**42.964**	151.832

The embedding of the new molecule is compared to known embeddings using built in function that calculates similarity score based on cosine similarity. Voting for fingerprint labels is then done proportionally based on similarity score. This approach, which corresponds to the weighted nearest neighbor, is further discussed in the section 5.

5 EVALUATION

We evaluated the learning methods using various metrics, with a focus on the most informative ones, such as hamming loss, label ranking loss, weighted F1 score, and coverage error [1], results of these evaluations are shown in Table 3. To ensure robust evaluation, we employed a 5-fold cross-validation approach, which we repeated twice to obtain reliable performance measurements.

Table 3: Random Forest performance metrics

	Default Classifier	Similarity Voting	Random Forest
Hamming Loss	0.083	0.038	0.043
Weighted F1 Score	0.635	0.642	0.854
Label Ranking Loss	0.630	0.083	0.010
Coverage Error	166.000	64.794	42.964

The Default Classifier always predicts the majority class for each label.

Similarity Voting uses Spec2Vec similarity to proportionally vote for labels. This approach is presented as a stronger baseline from which we can measure improvements of our models.

Random Forests were trained for each label, using One Vs Rest (OVR) method. Each forest had 100 estimators with balanced class weights (inversely proportional). Impurity was measured using Gini Impurity measure and no other restricting parameters were set - the defaults of sklearn Random Forest Classifier apply.

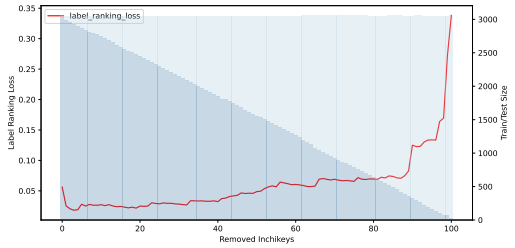


Figure 5: Models ability to generalize to unseen InChI Keys.

Our goal isn't predicting fingerprints for known molecules, but handling new ones effectively. To test this, we deliberately removed some InChI Keys from our dataset. By doing this, we checked how well our models perform in predicting the structures of these unfamiliar molecules. This real-world scenario testing helps us understand how practical and effective our approach is when dealing with novel compounds not present in our initial training data.

We have also performed 10-fold validation by removing 10 InChI Keys at a time from the training data. The model was trained on the remaining ~90 InChI Keys (~2700 samples of mass spectra) and evaluated on ~10 unseen ones (~300 samples of mass spectra). The results are shown in Table 5. The Random Forests' ability to predict larger amounts of unseen InChI Keys and effects of less training data and therefore less diverse embedding knowledge is shown in Figure 5. Even though the label ranking loss is increasing it is still well below the loss of the Default Classifier and even Similarity Voting, when a large amount of InChI Keys are missing and the training dataset is smaller.

Table 4: Similarity Voting on Unseen InChI Keys

	Hamming Loss	Weighted F1 Score	Label Ranking Loss	Coverage Error
average	**0.047**	**0.639**	**0.084**	**75.153**

Here only the average is shown to provide a reference point for the quality of Random Forests. More data was not included to not clutter the article. Unseen InChI Keys were simulated by keeping only the test rows (unseen InChI Keys) and train columns (other InChI Keys) in the similarity matrix.

Table 5: 10-fold evaluation results for unseen InChI Keys, Results per Fold

	Hamming Loss	Weighted F1 Score	Label Ranking Loss	Coverage Error
0	0.068	0.749	0.043	63.432
1	0.064	0.806	0.039	85.369
2	0.061	0.775	0.045	94.405
3	0.066	0.757	0.031	70.266
4	0.060	0.759	0.033	79.687
5	0.101	0.676	0.066	97.522
6	0.124	0.596	0.077	115.793
7	0.036	0.864	0.019	63.857
8	0.047	0.818	0.017	64.828
9	0.077	0.721	0.063	84.503
average	**0.070**	**0.752**	**0.043**	**81.966**

6 REPRODUCIBILITY

The whole pipeline and evaluation were built with repeatability in mind to allow for future studies, model comparisons, and reevaluation of results. The dataset used is public, Spec2Vec models are built upon these data, and model training functions along with parameters are available in the repository github.com/alpi314/mass_spectra tagged *article*. Training of the models is done with fixed random seeds and stores models with training parameters, train and test data with the use of the pickle package. Metrics and evaluations are always stored along with the models.

7 CONCLUSION

Our results demonstrate that Spec2Vec embeddings of TMS can effectively be converted into molecular fingerprints using machine learning methods. These methods have proven to be reliable even when predicting molecular structures for molecules that have not been encountered before. This is significant because it allows processing new MS spectra to uncover their most likely structural components, which we can then match against databases. This structural information can be directly applied in various research studies. Our plans for future work involve expanding this approach to larger compound databases. Additionally, we plan to broaden our research to predict more SMARTS patterns as part of expanding our molecular fingerprint prediction capabilities. While we'll stay focused on fingerprints for database queries, we will be also looking into predicting arbitrary SMARTS patterns.

REFERENCES

[1] Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. 2022. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203, 117215. DOI: 10.1016/j.eswa.2022.117215.

[2] Juliane Glüge, Kristopher McNeill, and Martin Scheringer. 2023. Getting the SMILES right: identifying inconsistent chemical identities in the ECHA database, PubChem and the CompTox Chemicals Dashboard. *Environmental Science: Advances*, 2, 4, 614. DOI: 10.1039/D2VA00225F.

[3] Florian Huber, Lars Ridder, Stefan Verhoeven, Jurriaan H. Spaaks, Faruk Diblen, Simon Rogers, and Justin J. J. van der Hooft. 2021. Spec2Vec: Improved mass spectral similarity scoring through learning of structural relationships. *PLOS Computational Biology*. DOI: 10.1371/journal.pcbi.1008724.

[4] Florian Huber, Stefan Verhoeven, Christiaan Meijer, and Hanno Spreeuw. 2020. matchms - processing and similarity evaluation of mass spectrometry data. *Journal of Open Source Software*, 5, 2411. DOI: 10.21105/joss.02411.

[5] Rontani Jean-Francois. 2022. Use of Gas Chromatography-Mass Spectrometry Techniques (GC-MS, GC-MS/MS and GC-QTOF) for the Characterization of Photooxidation and Autoxidation Products of Lipids of Autotrophic Organisms in Environmental Samples. *Molecules*, 27, 5. DOI: 10.3390/molecules27051629.

[6] Hiroyuki Kuwahara and Xin Gao. 2021. Analysis of the effects of related fingerprints on molecular similarity using an eigenvalue entropy approach. *Journal of Cheminformatics*, 13, 1, 27. DOI: 10.1186/s13321-021-00506-2.

[7] Milka Ljoncheva, Tina Kosjek, Sašo Džeroski, and Sintija Stevanoska. 2023. GC-EI-MS datasets of trimethylsilyl (TMS) and tert-butyl dimethylsilyl (TBDMS) derivatives. *Mendeley Data*. DOI: 10.17632/j3z5bmvmnd.6.

[8] Milka Ljoncheva, Tomaž Stepišnik, Tina Kosjek, and Sašo Džeroski. 2022. Machine learning for identification of silylated derivatives from mass spectra. *Journal of Cheminformatics*, 14, 1, 62. DOI: 10.1186/s13321-022-00636-1.

[9] Milka Ljoncheva, Sintija Stevanoska, Tina Kosjek, and Sašo Džeroski. 2023. GC-EI-MS datasets of trimethylsilyl (TMS) and tert-butyl dimethyl silyl (TBDMS) derivatives for development of machine learning-based compound identification approaches. *Data in Brief*, 48, 109138. DOI: 10.1016/j.dib.2023.109138.

[10] 2013. RDkit MACCS Keys. Accessed on 2023-08-31. (2013). https://github.com/rdkit/rdkit-orig/blob/master/rdkit/Chem/MACCSkeys.py.

[11] Egon L. Willighagen et al. 2017. The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of Cheminformatics*, 9, 1, 33. DOI: 10.1186/s13321-017-0220-4.

A meaty discussion: quantitative analysis of the Slovenian meat-related news corpus

Matej Martinc
Jožef Stefan Institute
Ljubljana, Slovenia
matej.martinc@ijs.si

Senja Pollak
Jožef Stefan Institute
Ljubljana, Slovenia
senja.pollak@ijs.si

Andreja Vezovnik
University of Ljubljana
Ljubljana, Slovenia
andreja.vezovnik@fdv.uni-lj.si

ABSTRACT

We conduct a quantitative analysis of the meat-related news in the Slovenian news media. As a first step, we construct a corpus containing news articles related to the topic of meat. Next, we conduct a topical and temporal analysis of the corpus using state-of-the-art natural language processing techniques for topic modeling and semantic change detection. The results show that economic topics related to meat, which have been prevailing more than a decade ago, are being replaced by cultural (especially culinary), ecological, and health topics. The results also indicate that there is a trend in Slovenian news coverage of framing veganism in relation to health and environment.

KEYWORDS

news analysis, topic modeling, semantic change detection

1 INTRODUCTION

In this study, we focus on the media coverage of a subject that is becoming more important due to its connection to the health and ecological issues of contemporary societies, meat. On one hand, meat is seen as a perfect nutritional pack, and its consumption is considered natural, normal, necessary, and enjoyable [10]. On the other hand, meat production heavily impacts the environment and can be seen as unhealthy and unsafe for human consumption [2]. These angles are reflected in news media debates, which lately showed a significant presence of anti-meat consumption and/or production narratives [9]. Several studies have also pointed out increased media coverage of veganism [7] and meat alternatives, especially cultured meat, produced by culturing animal cells in vitro [4].

While several studies explored different meat narratives in English news media [9, 4], analysis of meat narratives in the Slovenian news remains a research gap. To fill this gap, we conduct a quantitative analysis of how the concept of meat is presented in the Slovenian media and try to identify stable trends in the news about meat, in order to show how the notion of meat changed in Slovene news media over time. For the analysis, we employ state-of-the-art (SoA) natural language processing (NLP) techniques, which have proved themselves useful for analysis of social trends and topics in different languages. To identify main topics related to the concept of meat and to detect temporal trends concerning attitudes towards meat, we employ BERTopic [3], the current SoA approach for topic identification based on clustering of contextual embeddings, on the corpus of Slovenian news. To investigate changes in attitudes towards some specific meat related topics, we additionally employ a model for semantic change detection, which analyses temporal changes in usage of words [6].

This is the first quantitative analysis of Slovenian news articles that tries to automatically identify the main topics related to meat and how their popularity changes through time. We are also not aware of any studies, in which meat narratives would be analysed with NLP techniques.

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

2 METHODOLOGY

2.1 Dataset construction

In order to explore the Slovenian news media about meat, we first construct a corpus that would allow us to conduct a topical and temporal analysis of news articles about meat. To do that, we obtained news articles from a large news database from a Slovenian clipping agency. The obtained articles needed to contain one of the two words[1]: meso (meat) and živinoreja (animal husbandry). The final obtained corpus covers a period from 2008 until 2019[2] and was split into five distinct temporal chunks, each covering two years, for the purpose of temporal analysis. The corpus structure is presented in detail in Table 1.

The corpus contains articles from nine Slovenian news sources:

- three daily newspapers with long tradition, published online and in print, **Delo, Večer** and **Dnevnik**,
- the weekly issues of the publishers under item 1, **Delo - Sobotna priloga, Dnevnik - Dnevnikov objektiv, Večer - V soboto**, and **Večer v nedeljo**, published on the weekends,
- **24ur.com**, which is the most visited web news portal in Slovenia, and **Rtvslo.si** is a web news portal of the Slovenia's national public broadcasting organization.

2.2 Topical analysis

We propose a two step corpus analysis approach in order to determine the main topics emerging in relation to meat in the Slovenian news corpus and to explore how these topics change through time. In the first step, we use BERTopic [3] to determine the main topics in the corpus. It uses Sentence Transformers [11] to generate document representations. These representations are clustered using Hierarchical density based clustering (HDBSCAN) [8]. Finally, coherent topic representations are extracted by employing a class-based variation of a term frequency-inverse document frequency (TF-IDF). The resulting topic distribution across corpus obtained by BERTopic is different from the distribution obtained by conventional topic models, such as Latent Dirichlet allocation, since each document in the corpus only belongs to either **one** or **none** of the topics.

[1] Due to the morpohological richness of Slovenian, the search query did not cover only basic form of each word, but also several of its morphological derivatives.
[2] This time period was chosen due to the lack of available articles before the year 2008 and due to the COVID-19 pandemic, which had a drastic influence on the media focus and coverage in the time period 2020/2021.

Source	2008/2009	2010/2011	2012/2013	2014/2015	2016/2017	2018/2019	All
24ur.com	61	83	99	143	156	296	838
Delo	496	506	648	690	599	648	3587
Delo - Sobotna priloga	57	72	95	86	76	98	484
Dnevnik	360	405	697	725	630	805	3622
Dnevnik - Dnevnikov objektiv	44	63	71	71	76	114	439
Rtvslo.si	27	51	107	197	332	491	1205
Večer	445	406	768	678	520	614	3431
Večer - V soboto	23	50	86	105	82	108	454
Večer v nedeljo	0	0	0	226	290	286	802
All	1513	1636	2571	2921	2761	3460	14862

Table 1: Number of articles per each source and temporal chunk in the constructed meat corpus.

By not restricting the number of topics, the model returns 156 topics. The manual inspection revealed that most of these topics are too specific, i.e. describing just one or two specific meat related events that were covered in the Slovenian news. To solve this problem, we reduce the number of topics by iteratively merging the class-based TF-IDF representations of the least common topic with its most similar one, in order to obtain predefined number of k topics (see [3] for details). We set the k to 20, which represents a balanced trade-off between interpretability allowed by a small number of topic and specificity offered by a large number of topics.

The obtained topics were manually inspected and grouped into five manually defined categories related to the object of meat, according to the common thread pervasive across several topics. This manual grouping into larger categories (e.g. economic, ecology, ...) allows us to determine the relative importance of several "general" aspects of news covering meat in contemporary media landscape. It also allows us to focus our analysis just on the more interesting aspects of news on meat in the next step, i.e. aspects which show clear increasing/decreasing temporal trends.

2.3 Temporal analysis

To determine how the topic of meat changes over time, the corpus is split into temporal slices. We calculate topic distribution for each slice in order to obtain relative counts (i.e. the number of articles belonging to a single topic divided by the number of all articles published in a specific time slice that belong to any topic[3]) for each topic. This allows us to determine relative "importance" of a specific topic in a specific time period and enables us to identify increasing/decreasing trends for specific topics by visualizing how the relative importance changes across time. The same procedure is applied to determine relative "importance" and detect trends on the level of manually defined categories.

For topics, which show increasing coverage trend and are more interesting from a sociological point of view, we also conduct an additional temporal analysis, by employing a procedure similar to the one proposed by Martinc et al. [6], where the information from the set of contextual token embeddings is aggregated into temporal representations by averaging. More specifically, we use a Transformer language model to generate contextual token embeddings. Tokens that have the same lemma and appear in the same temporal chunk are averaged in order to obtain a temporal vector representation for a specific lemma. These vectorised temporal representations are used for a focused analysis of manually selected concepts (i.e., "meat" and "vegan") and their semantic correlation (measured with cosine distance between temporal representations) to words representing a specific topic.

While in Martinc et al. [6] temporal representations were generated for an entire corpus, in our approach we propose a filtering step based on the previous topic modeling step. BERTopic uses HDBSCAN for topic clustering, a soft-clustering approach that allows noise to be modeled as outliers. The authors claim that this prevents unrelated documents to be assigned to any of the topics and generally improves topic representation [3]. Since in our temporal analysis we are interested in historical trends, i.e. consistent changes through time that reflect cultural and social shifts in attitudes towards meat, we hypothesise that removing the outlier documents not belonging to coherent topics might allow us to conduct a more focused temporal analysis, which will only cover main topical trends and disregard semantic changes in word meaning that occur due to events covered in news that do not reflect broader cultural trends or narratives. For this reason, we filter out articles from the corpus not belonging to any topic and only generate temporal lemma representations on articles belonging to topics assigned by BERTopic.

3 EXPERIMENTS

3.1 Experimental setting

The experiments are conducted on the Slovenian news corpus described in Section 2.1. For topic modeling, we employ BERTopic with a multilingual embedding model, namely the "paraphrase-multilingual-MiniLM-L12-v2" Sentence transformer from the Huggingface library[4], since no monolingual Sentence transformer model exists for Slovenian. For generation of temporal representations, we employ the SloBERTa model [12]. As was mentioned in Section 2.3, the temporal representations are created by averaging token embeddings appearing in the same time slice and having the same lemma. To obtain the lemmas, we label the entire corpus with the Classla lemmatizer [5].

3.2 Results

The English translation of topics obtained are presented in Table 2. 9,335 articles were labeled as not belonging to any specific topic. Among the categorized articles, most were categorised in the **topic** "restaurant, wine, kitchen, meat, culinary", which contains 745 articles describing Slovenian gastronomy. The smallest were the topics containing articles about the influence of meat industry on the environment, public health, and veganism, each of these topics containing just about 100 articles.

Manual inspection of different topics revealed that several topics can be further aggregated into broader **categories**, due to

[3]Articles classified as not belonging to any topic, are disregarded in the calculation of relative counts.

[4]https://huggingface.co/

Category	Translated topic	Count
economy	percentage, inflation, price increase, chicken, food	228
economy	euro, ljubljana, million, company	202
economy	bank, mip, euro, million, supervisory	125
economy	slovenian, food, quality, consumer, percentage	646
economy	slovenian, company, mercator, euro, million	204
culture	book, other, write, story, time	148
culture	show, theatre, director, festival, theatrical	207
culture	tourism, time, old, big, house	336
culture	restaurant, wine, kitchen, meat, culinary	745
ecology and health	vegan, child, animal, veganism	114
ecology and health	water, dioxide, greenhouse, carbon, energy	104
ecology and health	fat, cholesterol, diet, food, health	138
ecology and health	marine, whaling, dolphin, fish, allowed	114
agriculture	milk, agriculture, percentage, organic, Slovenian	239
agriculture	meat, kebab, horse, product, dioxin	319
other	other, can, life, time, world	429
other	coach, team, season, play, championship	346
other	oil, meat, minute, water, paprika	299
other	prison, police officer, prosecution, convicted, euro	201
other	election, president, agreement, government, political	383
not categorized	/	9335

Table 2: Topics and manually defined categories in the Slovenian meat corpus.

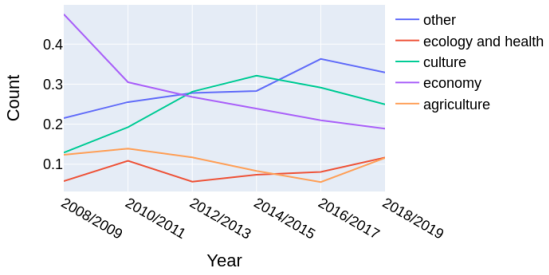


Figure 1: Category distribution across time.

the fact that several topics covered semantically similar content (e.g., topics "euro, ljubljana, million, company" and "bank, mip, euro, million, supervisory" both include financial news about different Slovenian meat companies). More specifically, the topics were manually categorized as: *"economy", "culture", "ecology and health", "agriculture"*, and category *"other"*, containing articles covering several topics with very different content that can not be combined into a broader semantic category, such as sport, life style, recipes, politics, and judiciary. Ignoring the category named "other", most articles covered economy and culture. These categories were identified based on previous sociological research on meat [13]. By combining some topics into broader categories, besides temporal analysis of somewhat specific topics, we are also able to conduct temporal analysis on a more general level that might allow us to detect how distinct general aspects of the meat related news loose or gain in popularity through time. Figure 1 shows the distribution of categories across time.

While economic topics were the most prevailing in 2008/2009, a graph also shows a clear decreasing trend of this category occurred after 2010. The most upward trend is in the amount of articles belonging to the category "other", which becomes the most dominant in 2016/2017. The production of articles covering cultural topics has also been steeply increasing until 2014/2015, after that a gradual decline is observed. While agricultural topics do not indicate any clear positive or negative trends throughout the years, the ecology and health topics appear to be gaining in popularity in the recent years, especially from 2012/2013 forward.

Figure 2 shows relative counts (i.e. the number of articles belonging to specific topics divided by all articles that were assigned a topic) for topics inside a specific category. Using this fine-grained view, one can see that the rise in *culture*-related topics can be contributed to the major increase in the amount of articles belonging to the topic "restaurant, wine, kitchen, meat, culinary" in 2012/2013, which mostly covers Slovenian gastronomy.

When it comes to *economic* topics, we can see that all but one topic (i.e. the topic "slovenian, food, quality, consumer, percentage", which differs from other economic topics by being more focused on the quality/price ratio) in this category decline in terms of relative count significantly in 2010/2011.

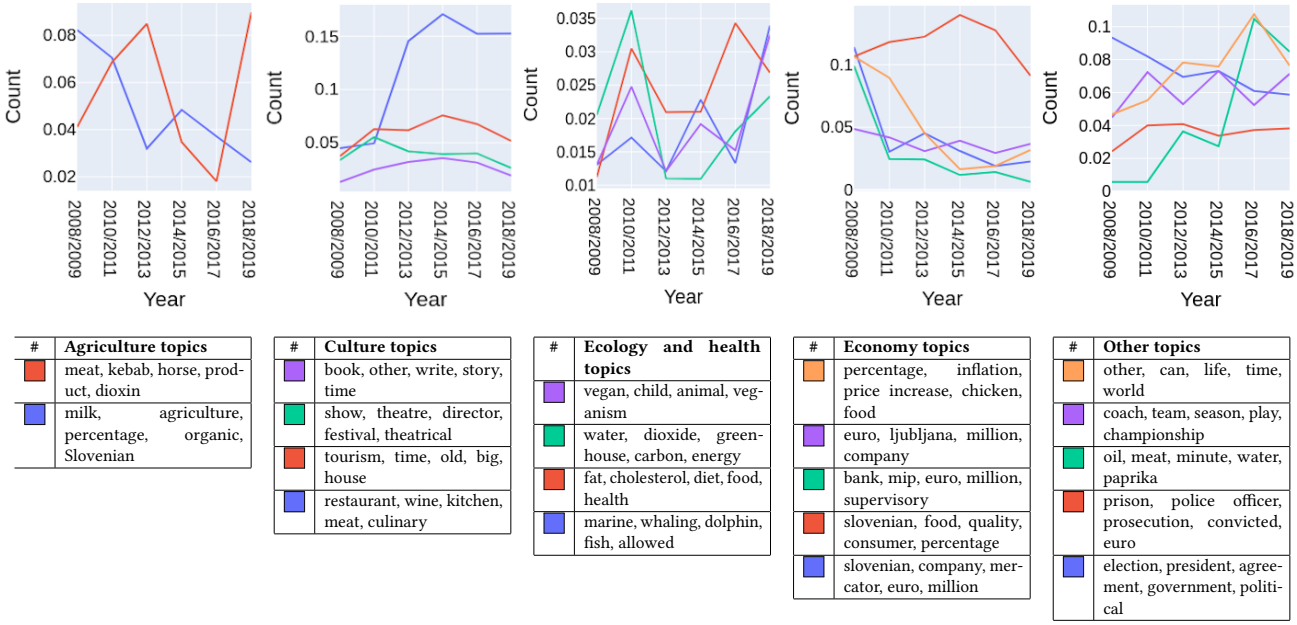
In the *ecology and health* category, one can see an increase in the relative count of topics covering veganism and over-fishing. While the popularity of the topic covering health benefits and drawbacks of meat is also increasing, the environmental topics related to global warming have decreased in popularity from the peak in 2010/2011. In the *agriculture* category, we see clear peaks in discussion on the topic "meat, kebab, horse, product, dioxin", which includes coverage of some scandals related to meat production and products in specific years. The topic most responsible for the increasing trend in the "*other*" category is "oil, meat, minute, water, paprika", which mostly covers articles about food recipes.

Finally, we discuss results of the focused temporal analysis for two manually selected concepts, "meat" and "vegan" (see Figure 3). We decided to explore an aspect of meat related to creation of cultured meat (meat produced from animal stem cells) and plant based meat analogues, which was not detected in our automatic topic analysis due to the scarcity of journalistic articles addressing cultured meat, but was nevertheless addressed by several scholars studying media representation of cultured meat [1]. We looked into semantic similarity between words "meat" and words "artificial", "laboratory", and "substitute". One can see that the cosine similarity between "meat" and all related concepts peaks in 2012/2013. This coincides with the development of cultured meat and plant-based meat analogues and the consequential news reporting on it. The first public tasting of cultured burger occurred in 2013 in London. After 2012/13, only the cosine similarity between "substitute" and "meat" keeps increasing, while we see a trend of stagnation or even gradual decrease in semantic similarity for the other two concepts. This suggests that the Slovenian news media is not significantly expanding the coverage of production of the artificial meat in recent years.

Due to the findings of the automated temporal topic analysis, suggesting a constant growth in popularity of the topic covering veganism, we also opted for a further analysis of the word "vegan". We were interested how the concept is correlated with words "healthy", "environment", "ecological", and "climate change" in order to test the hypothesis that the news media is more and more connecting veganism to ecological and health related issues. The results indicate a stable positive trend throughout the years in terms of cosine similarity between veganism and selected concepts, confirming our hypothesis.

4 CONCLUSION

In this study, we have conducted a quantitative analysis of the meat related news in Slovenian news media. We constructed a corpus of meat related news articles and conducted topical and temporal analysis of the corpus using several SoA NLP techniques. We identified the main meat-related topics and trends and detected which meat related topics are gaining/loosing media coverage and popularity.



#	Agriculture topics
	meat, kebab, horse, product, dioxin
	milk, agriculture, percentage, organic, Slovenian

#	Culture topics
	book, other, write, story, time
	show, theatre, director, festival, theatrical
	tourism, time, old, big, house
	restaurant, wine, kitchen, meat, culinary

#	Ecology and health topics
	vegan, child, animal, veganism
	water, dioxide, greenhouse, carbon, energy
	fat, cholesterol, diet, food, health
	marine, whaling, dolphin, fish, allowed

#	Economy topics
	percentage, inflation, price increase, chicken, food
	euro, ljubljana, million, company
	bank, mip, euro, million, supervisory
	slovenian, food, quality, consumer, percentage
	slovenian, company, mercator, euro, million

#	Other topics
	other, can, life, time, world
	coach, team, season, play, championship
	oil, meat, minute, water, paprika
	prison, police officer, prosecution, convicted, euro
	election, president, agreement, government, political

Figure 2: Relative counts for topics "agriculture", "culture", "ecology and health", "economy", and "other".

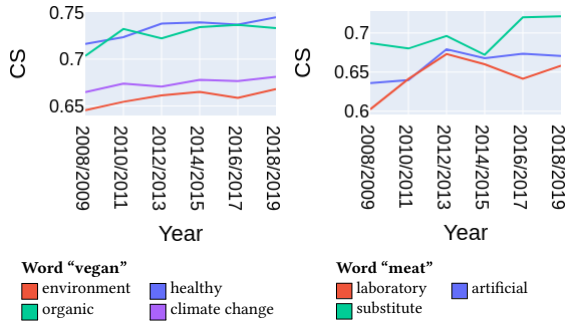


Figure 3: Cosine similarity (CS) between the words "vegan" (left) and "meat" (right), and selected concepts.

The results indicate that topics related to the meat economy are loosing ground to cultural (especially culinary), ecological, and health topics. On the other hand, agricultural topics are not gaining/loosing news coverage across time. The topic of artificial meat is not yet carefully covered in Slovenian media and since the initial increase in coverage in 2012/2013 has not been gaining further traction. On the other hand, the results show that there is semantic relation between the words vegan, healthy, and ecological, which is also slowly increasing over time.

In the future, we will further explore main developments of the meat narrative in Slovenian media by gathering a larger corpus covering more media sources, which will allow us to employ other approaches for topic analysis and semantic change detection that require more data. We will also explore other concepts and discourses in Slovenian media besides meat, such as immigration, using techniques similar to the ones proposed in this work. Finally, we plan to expand the analysis to also cover media reporting in neighboring countries.

5 ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency for research core funding for the programmes Knowledge Technologies (No. P2-0103) and the project Computer-assisted multilingual news discourse analysis with contextual embeddings (No. J6-2581).

REFERENCES

[1] Sghaier Chriki, Marie-Pierre Ellies-Oury, Dominique Fournier, Jingjing Liu, and Jean-François Hocquette. 2020. Analysis of scientific and press articles related to cultured meat for a better understanding of its perception. *Frontiers in psychology*, 11, 1845.

[2] International Agency for Research on Cancer et al. 2015. Iarc monographs evaluate consumption of red meat and processed meat. *World Health Organization. http://www. iarc. fr/en/mediacentre/pr/2015/pdfs/pr240_E. pdf.*

[3] Maarten Grootendorst. 2022. Bertopic: neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

[4] Patrick D Hopkins. 2015. Cultured meat in western media: the disproportionate coverage of vegetarian reactions, demographic realities, and implications for cultured meat marketing. *Journal of Integrative Agriculture*, 14, 2, 264–272.

[5] Nikola Ljubešić and Vanja Štefanec. 2020. The CLASSLA-StanfordNLP model for lemmatisation of non-standard serbian 1.1. Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1351, (2020).

[6] Matej Martinc, Petra Kralj Novak, and Senja Pollak. 2020. Leveraging contextual embeddings for detecting diachronic semantic shift. English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, (May 2020), 4811–4819. ISBN: 979-10-95546-34-4.

[7] Helen Masterman-Smith, Angela T Ragusa, and Andrea Crampton. 2014. Reproducing speciesism: a content analysis of australian media representations of veganism. In *Proceedings of the Australian Sociological Association Conference*.

[8] Leland McInnes, John Healy, and Steve Astels. 2017. Hdbscan: hierarchical density based clustering. *J. Open Source Softw.*, 2, 11, 205.

[9] Gilly Mroz and James Painter. 2022. What do consumers read about meat? an analysis of media representations of the meat-environment relationship found in popular online news sites in the uk. *Environmental Communication*, 1–18.

[10] Jared Piazza, Matthew B Ruby, Steve Loughnan, Mischel Luong, Juliana Kulik, Hanne M Watkins, and Mirra Seigerman. 2015. Rationalizing meat consumption. the 4ns. *Appetite*, 91, 114–128.

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, (Nov. 2019).

[12] Matej Ulčar and Marko Robnik-Šikonja. 2021. Sloberta: slovene monolingual large pretrained masked language model.

[13] Andreja Vezovnik and Tanja Kamin. 2020. Good food for the future: an exploration of biocapitalist transformation of meat systems. *Discourse, Context & Media*, 33, 100354.

Slovene Word Sense Disambiguation using Transfer Learning

Zoran Fijavž
University of Ljubljana, Faculty of Education
Slovenia
zoran.fijavzz@gmail.com

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

ABSTRACT

Word sense disambiguation is an important task in natural language processing and computational linguistics with several practical applications, such as machine translation and speech synthesis. While the bulk of research efforts are targeted to English, some multilingual resources which include Slovenian have emerged recently. We utilized the Elexis-WSD dataset and a multilingual large language model to train models for word sense disambiguation in Slovenian, using sentence pairs with matching lemmas and matching or different word senses. The best model achieved an F_1 score of 81.6 on a Slovenian test set, although the latter had a restricted vocabulary due to filtering and is not comparable other testing frameworks. The exhaustive generation of sentence pairs for given lemmas and senses did not improve model performance and reduced the performance in out-of-vocabulary testing. Training on a mixed English-Slovene dataset maintained high test set as well as out-of-vocabulary results.

KEYWORDS

word sense disambiguation, transfer learning, multilingual transformer

1 INTRODUCTION

Word sense disambiguation (WSD) aims to identify the correct word sense used in a particular context. It is a long-standing problem in the field of computational linguistics and is important for downstream applications, such as machine translation, information retrieval, text mining, and speech synthesis. Recent WSD approaches use pre-trained large language models such as BERT [3], fine-tuning them on annotated data. As with most supervised machine learning approaches, there is a bottleneck on high-quality training data acquisition. The problem is severe, as standard WSD approaches treat each word sense as a separate target label. A partial solution is to use multilingual pretrained models that can leverage several WSD datasets.

In this paper, we demonstrate a methodology for cross-lingual transfer learning for WSD in Slovene that does not require compatible sense inventories in different languages. The proposed approach also works on out-of-vocabulary data.

After outlining related works in Section 2, we describe WSD models we developed for Slovene in Section 3, and their evaluation in Section 4. In Section 5, we provide an interdisciplinary critique of the current approaches to WSD that may be informative for future research. Section 6 presents the conclusions and ideas for further work.

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

2 RELATED WORK

One of the first WSD algorithms was Lesk [11] and its various extensions that are based on the word overlap between pre-defined sense definitions and target sentences. Conceptually, modern approaches to WSD remain strikingly similar, with advances stemming mostly from increasingly complex word representations (e.g. contextual word embeddings) and expansive lexicographical resources (e.g. a gloss list for word senses in SemCor). Recent approaches use supervised learning directly on word sense annotations [5], enrich sense definitions with various lexicographical resources [7, 19] and include lexical databases as graph data in conjunction with contextual word embeddings [2].

Until recently, the development of contemporary WSD models for Slovenian has been hindered by a lack of available datasets. That was partly addressed by the inclusion of Slovenian in the multilingual Elexis-WSD and XL-WSD datasets [12, 16]. Models trained on the latter obtained an F_1 score of 68.36% for Slovene WSD, which is significantly lower than state-of-the-art English models scoring 80% or above (although differing test frameworks preclude direct comparisons).

3 METHODOLOGY

In this section we describe the training procedure, data preparation and testing framework used to develop and test the Slovenian WSD models.

3.1 Training Task and Setup

We operationalized WSD as a sentence-pair binary classification task that distinguishes between sentence pairs with an identical or distinct word sense for a target lemma. Word senses were thus defined solely through annotated examples without the need for a secondary source of sense definitions (e.g. sense collocations, coarse semantic tags or glosses). Casting WSD as a binary classification task allowed us to combine Slovene and English datasets, as sentence pairs could be generated from different WSD datasets irrespective of sense inventory compatibility. Examples of the sentence pairs can be found in Table 1. The drawback of this approach was a significant data loss from filtering, as many lemmas did not have enough senses and use examples to generate sentence pairs.

For the base model, we used the pre-trained model CroSloEngual BERT [22] that can encode Slovenian, Croatian, and English texts. To reduce the training time and computational requirements, we used bottom layer freezing [10], gradient accumulation, and early stopping for non-converging models. Hyperparameter tuning was done on a 10% sample of the training data. We set the learning rate to 3e-5, gradient accumulation steps to 16, the batch size to 48, and the number of epochs to 2. Training a single model on 20% of all Slovenian sentence pairs required approximately 4 hours using a 16 GB NVidia GPU.

Table 1: Two Examples of the lemma *Cirkus* in the Pair Dataset and its English translation.

Lemma	Sentence 1	Sentence 2	Match
Cirkus	Družina na sliki s 'cirkusom' postuje po deželi.	Uprava 'cirkusa' ni odpovedala predstave.	Yes
Circus	*Family on the photo travels around the country with 'circus'.*	*The 'circus' management did not cancel the show.*	*Yes*
Cirkus	Uprava 'cirkusa' ni odpovedala predstave.	Zganjali so 'cirkus' okrog družinskih vrednot.	No
Circus	*The 'circus' management did not cancel the show.*	*They were making 'circus' around family values.*	*No*

Table 2: Number of Sentences, Lemmas and Word Senses in Datasets.

Datasets	Sentences (n)	Lemmas (n)	Word senses (n)
Original Sl.	202,240	5,604	11,069
Filtered Sl.	139,445	1,597	4,633
Full Sl. train	104,316	1,597	4,633
10% Sl. train	99,205	1,597	4,633
20% Sl. train	102,548	1,597	4,633
Validation	6,972	691	1,743
Test	28,157	1,597	4,633
10% En. train	27,028	2,852	9,683
20% En. train	27,123	2,852	9,683
20% mix train	126,233	4,437	14,316
OOV	3,006	25	50

3.2 Data Preparation

We used both Slovenian and English WSD datasets. The Slovenian data was obtained from the Slovenian section of the Elexis-WSD corpus [12] and the English data was drawn from SemCor to approximately match the size of the filtered Slovenian data.

Over 50% of the original Slovenian lemmas had a single sense tag. We removed multi-word and hyphenated senses and repeatedly filtered the datasets until there were at least two senses per lemma with at least four examples. The original dataset was thus heavily filtered from 202,240 sentences with 5,604 lemmas and 11,069 word sense tags to 139,445 sentences with 1,597 lemmas and 4,633 word sense tags. Punctuation was removed and target words were enclosed in apostrophes as a weak supervision signal [7].

The filtered Slovenian dataset was split into train, test and validation datasets. For the test dataset, we sampled two or eight sentences per word sense (depending on the total number of available sentences). The lower limit was needed to create sentence pairs and the upper limit was used to prevent frequent lemmas and senses from giving overly optimistic test scores. The validation dataset was created by sampling four sentences per word sense from lemmas with at least eight sentences, assuming frequent senses would be sufficient to detect over- and underfitting. The remainder of the data was kept for training. The Slovenian training and testing datasets contained the full coverage of included word Slovenian senses (4,633 distinct senses) and the validation dataset contained 1,743 senses. All Slovenian datasets included the full coverage of included lemmas (1,597). The Slovenian training dataset contained 104,316 unique sentences, the testing set 28,159 sentences and the validation dataset 6,972 sentences.

The filtered Slovene datasets were transformed into a dataset of sentence pairs by generating sentence combinations between sentences sharing a lemma. We limited the number of non-matching combinations generated to the number of possible matching combinations for each word sense. By storing infrequent sense pairs and downsampling frequent ones, we created two smaller Slovene sentence-pair datasets with the size of 10% and 20% of the original dataset.

The English dataset was created to complement the Slovenian one: we filtered out senses and lemmas that could not generate sentence pairs, filtered out infrequent lemmas, created a sentence-pair dataset and downsampled it to the size of the two smaller Slovenian datasets. The number of negative and positive pairs was roughly balanced for all pair datasets. Additionally, multiple smaller Slovene datasets [4, 13, 14, 17, 20, 21] were joined and filtered to create an out-of-vocabulary (OOV) dataset that included only lemmas absent from the main Slovenian dataset. The OOV dataset consisted of sentence pairs with matching or non-matching word senses for a target word. Table 2 summarizes the number of sentences, lemmas, and senses for each dataset.

In total, we trained 7 models that differed in the training data used: the entire Slovene dataset, the 10% Slovene dataset, the 20% Slovene dataset, the 10% English dataset, the 20% English dataset (with and without early stopping) and the mixed 20% dataset (a concatenation of the 10% Slovene and English datasets).

3.3 Evaluation Settings

Model performance was measured using the F_1 score and the Matthews correlation coefficient (MCC). The latter is a chi-square statistic computed from the confusion matrix of classification results. It served as an additional performance metric and enabled us to compare models without having to predict specific word sense tags (e.g., evaluate models on the OOV dataset with dissimilar lemmas and sense tags).

Two methods were used to predict the sense classes on the Slovenian test set. The first prediction method, called *the average sense probability heuristic* (ASP) used the test set structure with the models' binary classifier to determine the most likely sense. The target sentence was combined with all other test sentences sharing a lemma (except with itself) and a softmax value was obtained for each pair. The softmax values were averaged based on the sense tag of the non-target sentence and the sense with the highest average score was chosen as the sense prediction for the target sentence. The second prediction method used nearest neighbour matching between target sentence embeddings and *sense embeddings*. The latter were created by converting the entire Slovenian training and validation dataset into sentence embeddings [18] and averaging them by their word sense tags. The test sentences were likewise embedded and their sense label was predicted by selecting the sense embedding with the lowest cosine distance from the target sentence embedding.

The most frequent sense (MFS) heuristic as well as the sense embedding predictions from an untrained model were used as performance baselines. Lastly, several F_1 scores per model (micro-F_1, macro-F_1 and micro-F_1 by POS tags) were used as repeated

Table 3: F_1 Scores of Binary Classifier Predictions.

Model	Micro-F_1
MFS baseline	40.4
Full Sl.	81.0
10% Sl.	81.4
20% Sl.	80.5
10% En.	68.7
20% En.	46.9
20% En. (early stopping)	80.6
20% mix	**81.6**

Table 4: Binary Classifier MCC Test and OOV Scores.

Model	MCC test	MCC OOV
Full Sl.	**0.629**	0.273
10% Sl.	0.55	0.292
20% Sl.	0.578	0.284
10% En.	0.321	0.268
20% En.	0.004	0.273
20% En. (early stopping)	0.491	**0.353**
20% mix	0.578	0.326

Table 5: F_1 Scores of Nearest Neighbour Predictions.

Model	Micro-F_1
MFS baseline	40.4
Untrained model	21.7
Full Sl.	**72.8**
10% Sl.	50.9
20% Sl.	60.7
10% En.	53.2
20% En.	60.6
20% En. (early stopping)	28.7
20% mix	61.0

measures for model comparison using the Friedman test with the Nemenyi post-hoc test.

4 RESULTS

We evaluated model predictions with binary classifiers and with nearest neighbour matching to sense embeddings. Additionally, we used the Matthews correlation coefficient to evaluate the performance of binary classifiers and evaluate model performance on the out-of-vocabulary dataset.

4.1 Binary Classifier Sense Predictions

The baseline F_1 from the MFS heuristic was 40.4%. The difference between model predictions was statistically significant (χ^2_F = 36.12; df = 5; n = 8; p < 0.001) with the top three models differing significantly from the MFS baseline: the models, trained on the mixed 20% training data (F_1 = 81.6; p = 0.001), the 10% Slovene data (F_1 = 81.4; p = 0.026), the entire Slovene dataset (F_1 = 81; p = 0.004). Detailed results from predictions with binary classifiers can be found in Table 3. The statistical differences between binary classification models are presented in Figure 1.

4.2 Binary Classifier Correlation Metrics

As the testing set was transformable into sentence pairs, we used the binary classifiers directly on the test set and computed a MCC without predicting sense labels. We also applied the same procedure to test model performance on the OOV dataset.

The highest correlation between actual and predicted binary labels was achieved by the model, trained on the entire Slovenian dataset (MCC = 0.629) followed by models, trained on the 20% Slovene and 20% mixed datasets (MCC = 0.578; for both). The highest correlation between the actual and predicted labels on the OOV dataset was achieved by the model, trained on the 20% English dataset with early stopping (MCC = 0.353), followed by the 20% mixed dataset (MCC = 0.326). It should be noted that the former was a base model with minimal updates, as the training stopped after a single update at 200 out of 1916 total steps. Interestingly, ranking the models by the amount of included training data revealed a positive correlation between the number of included examples and the testing dataset MCC (r_s = 0.566; df = 5; p = 0.185) and a negative correlation between the number of included examples and OOV dataset MCC (r_s = -0.378; df = 5; p = 0.404), although neither association was statistically significant. Detailed results from MCC testing can be found in Table 4.

4.3 Sense Predictions with Nearest Neighbour Matching

For predictions with nearest neighbour matching between target sentence and sense embeddings, the baselines used were the MFS heuristic (F_1 = 40.4%) and the predictions from the untrained model (F_1 = 21.7%). The difference between model predictions was statistically significant (χ^2_F = 45.11; df = 5; n = 9; p < 0.001). The only model significantly different from the MFS predictions was trained on the entire Slovene dataset (F_1 = 72.8%; p = 0.003). Detailed results from predictions using nearest neighbour matching can be found in Table 5. The statistical differences between nearest neighbour predictions from different models are presented in Figure 2.

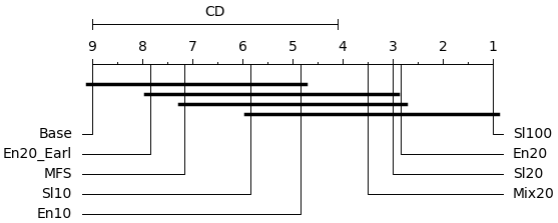


Figure 1: Critical Distance Diagram for Nearest Neighbour Results.

5 DISCUSSION ON INTERDISCIPLINARY ASPECTS

In this section, we offer a brief critique of the WSD task from the perspective of psycholinguistics, pragmatics and insights gained through model development, and suggest options for further research.

The datasets commonly used for WSD are not transparent in terms of the specific sense ambiguities they contain in spite of available typologies. Psycholinguistic literature has identified significant differences in human processing between homonymy

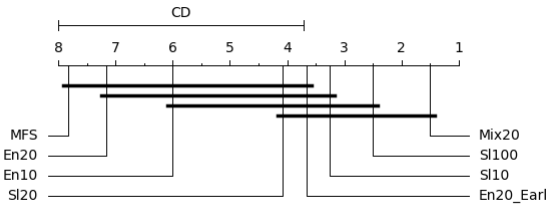


Figure 2: Critical Distance Diagram for Binary Classification Results.

and polysemy [8], as well as between various subtypes of the latter (e.g., metonymy and metaphors) [9]. As demonstrated by the use of the out-of-vocabulary test set, additional datasets, even if comparatively small, can provide important additional information on model performance. Incorporating a theoretically informed typology of polysemy or lexical ambiguity, future research could provide richer descriptions of word sense relations contained in widely used WSD datasets as well as develop specific tests for various types of polysemy. The latter could draw on datasets from psycholinguistic experiments, which commonly control for a plethora of variables, such as word and sense frequency. We also observed Elexis-WSD and SemCor contained a large number of single-sense lemmas, which would explain why F_1 scores from the MFS heuristic in related works are commonly relatively high.

Furthermore, while large language models have achieved state-of-the-art results in WSD, they do not fundamentally diverge from distributional semantics [6], which is but one account of possible disambiguation mechanisms. It is possible, for instance, to conceptualise word disambiguation as a pragmatic process whereby the common ground (shared knowledge) between speakers [1] scaffolds disambiguation and by which account speakers may introduce ambiguity on purpose to meet various communicative goals [15].

6 CONCLUSION

We developed several word sense disambiguation models for Slovenian text and achieved comparatively high performance, albeit on a limited selection of lemmas and word senses. We demonstrated that including small datasets to measure out-of-vocabulary performance yields important insights, as the models tended to generalize better with compacter training datasets.

The models presented in this paper would benefit from a review of Slovenian lexicographical sources and sense inventory compatibility between them. Replacing annotated sentences with sense definitions (e.g. collocation lists, coarse semantic tags, gloss definitions) would greatly increase the number of available training examples. Other large language models could also be used and a detailed hyperparameter optimization could be performed for each model individually.

The source code related to this paper and the datasets used are freely available[1].

Acknowledgments

The work was partially supported by the Slovenian Research and Innovation Agency (ARIS) core research programme P6-0411, and projects J6-2581 and J7-3159.

[1]https://github.com/zo-fi/slo_wsd_ZFMA

REFERENCES

[1] Keith Allan. 2013. What is Common Ground? In *Perspectives on Linguistic Pragmatics*. Perspectives in Pragmatics, Philosophy & Psychology. Alessandro Capone, Franco Lo Piparo, and Marco Carapezza, editors. Springer, Cham, 285–310. DOI: 10.1007/978-3-319-01014-4_11.

[2] Michele Bevilacqua and Roberto Navigli. 2020. Breaking Through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2854–2864. DOI: 10.18653/v1/2020.acl-main.255.

[3] 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)*, 4171–4186. DOI: 10.18653/v1/N19-1423.

[4] Zala Erič, Miha Debenjak, and Denis Derenda Cizel. 2022. Cross-lingual sense disambiguation. GitHub repository. https://github.com/dextos658/Cross-lingual-sense-disambiguation.

[5] Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5297–5306. DOI: 10.18653/v1/D19-1533.

[6] Zellig S. Harris. 1954. Distributional Structure. *WORD*, 10, 2-3, 146–162. DOI: 10.1080/00437956.1954.11659520.

[7] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3509–3514. DOI: 10.18653/v1/D19-1355.

[8] Ekaterini Klepousniotou and Shari R. Baum. 2007. Disambiguating the ambiguity advantage effect in word recognition: An advantage for polysemous but not homonymous words. *Journal of Neurolinguistics*, 20, 1, 1–24. DOI: 10.1016/j.jneuroling.2006.02.001.

[9] Ekaterini Klepousniotou, G. Bruce Pike, Karsten Steinhauer, and Vincent Gracco. 2012. Not all ambiguous words are created equal: An EEG investigation of homonymy and polysemy. *Brain and Language*, 123, 1, 11–21. DOI: 10.1016/j.bandl.2012.06.007.

[10] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4365–4374.

[11] Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation* (SIGDOC '86), 24–26. ISBN: 978-0-89791-224-2. DOI: 10.1145/318723.318728.

[12] Federico Martelli et al. 2022. Parallel sense-annotated corpus ELEXIS-WSD 1.0. *https://elex.is/*. Retrieved Oct. 21, 2022 from https://www.clarin.si/repository/xmlui/handle/11356/1674.

[13] Matej Miočić, Marko Ivanovski, and Matej Kalc. 2022. NLP-tripleM. GitHub repository. https://github.com/KalcMatej99/NLP-tripleM.

[14] David Miškić, Kim Ana Badovinac, and Sabina Matjašič. 2022. cross-lingual-sense-disambiguation. GitHub repository. https://github.com/NLP-disambiguation/cross-lingual-sense-%20disambiguation.

[15] Brigitte Nerlich and David D. Clarke. 2001. Ambiguities we live by: towards a pragmatics of polysemy. *Journal of Pragmatics*, 33, 1, (Jan. 2001), 1–20. DOI: 10.1016/S0378-2166(99)00132-0.

[16] Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Xl-wsd: an extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35, 15, 13648–13656. DOI: 10.1609/aaai.v35i15.17609.

[17] Erazem Pušnik, Rok Miklavčič, and Aljaž Šmaljcelj. 2022. nlp-project3. GitHub repository. https://github.com/RoKKim/nlp-project3.

[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. DOI: 10.18653/v1/D19-1410.

[19] Yang Song, Xin Cai Ong, Hwee Tou Ng, and Qian Lin. 2021. Improved Word Sense Disambiguation with Enhanced Sense Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 4311–4320. DOI: 10.18653/v1/2021.findings-emnlp.365.

[20] Jure Tič, Nejc Velikonja, and Sandra Vizlar. 2022. NLP. GitHub repository. https://github.com/JureTic/NLP.

[21] Andrej Tomažin. 2022. nlp-wic. GitHub repository. https://github.com/anzetomazin/nlp-wic.

[22] Matej Ulčar and Marko Robnik-Šikonja. 2020. FinEst BERT and CroSloEngual BERT. In *Text, Speech, and Dialogue*, 104–111. DOI: 10.1007/978-3-030-58323-1_11.

Predicting the FTSO consensus price

Filip Koprivec
filip.koprivec@ijs.si
JSI, FMF, AFLabs
Ljubljana, Slovenia

Tjaž Eržen
erzen.tjaz@gmail.com
AFLabs
Ljubljana, Slovenia

Urban Mežnar
urban.meznar@aflabs.si
AFLabs
Ljubljana, Slovenia

ABSTRACT

The paper presents a system for predicting cryptocurrency consensus prices within the Flare Time Series Oracle (FTSO), a decentralized oracle solution running on Flare blockchain. By leveraging a combination of smoothing techniques and machine learning methodologies, we detail and analyze the construction and performance of our own provider. This paper presents the FTSO mechanism, and basic information about the game theoretic background together with rewarding and submission protocol. Lastly, we present our provider's prediction accuracy for each coin.

KEYWORDS

FTSO, schelling point, machine learning, regression, smoothings

1 INTRODUCTION

The blockchain and decentralized finance (DeFi) sectors have seen significant growth, but they share a common challenge: securely accessing data not directly included in transaction signatures. This issue, known as the *oracle problem* [3], hinders the broader adoption of blockchain technologies as it's typically difficult to obtain reliable off-chain data. While various on-chain protocols offer solutions, each has its trade-offs concerning security, accuracy, and data reliability. Traditional centralized oracles present risks like data manipulation, whereas fully decentralized alternatives often suffer from latency and higher costs.

This paper examines the Flare Time Series Oracle, a decentralized oracle that uses a schelling point mechanism to aggregate data from multiple providers [11]. Fata providers submit price estimates every three minutes, with the system price determined as a weighted median of these submissions. Given the inherent price variability across exchanges and the indeterminate nature of asset prices within a three-minute window, there isn't a singular "correct" price. Providers aim to select a price close to the final median, incentivized by the reward system. This competitive environment, involving around 100 data providers, has shown resilience against market anomalies and exchange issues. The paper investigates machine learning techniques to predict this final median price using exchange data. Given the dynamic nature of the competition, our prediction methods are designed for adaptability.

2 RELATED WORK

While no literature precisely addresses the Flare FTSO, the general oracle problem has been extensively studied. Caldarelli [4] highlights the challenges of the blockchain oracle problem. Ellul [7] delves into its role in decentralized finance. Zohar and Eyal [15] provide a comprehensive study, while Caldarelli's subsequent work [2] offers an overview of oracle research. Liu et al. [14] survey various oracle implementation techniques. Notably, Alagha [1] introduces a reinforcement learning model to enhance oracle reliability [11].

The main oracle solution provider is Chainlink, which addresses the oracle problem with enhanced security and scalability in Chainlink 2.0 [5]. Zhang et al. [13] also detail their approach, providing insights for evolving projects like Flare FTSO in the oracle domain.

3 FTSO PROTOCOL

The Flare Time Series Oracle plays an important role in Flare Network's data accuracy and decentralization. The protocol works in a series of discrete steps to decrease the performance hit on the whole network. Every 3 minutes marks the beginning of a new *price epoch*. Providers are mandated to submit their price estimates in a timely manner using the commit and reveal scheme to maintain confidentiality and prevent other providers from viewing or copying their predictions.

Only after the price epoch has ended, providers reveal the actual submitted values. This reveal must be done in the first 90 seconds of the next price epoch, which overlaps with the first half of the next submit epoch. After the reveal epoch ends, all the revealed values are combined and a network-wide price is calculated. Data providers are incentivized to submit *good* prices by the network-wide rewarding system, by being rewarded if prices fall in the middle two quartiles (IQR range) of the final price.

The network thus gets fresh asset prices every 3 minutes with some delay due to the reveal period. Such data granularity is not sufficient for high-frequency trading but has proven sufficient for many financial applications. The network and community explicitly don't define what a correct price is, to remove the vulnerability of the definition relying on a specific price source. Assets are denominated in $ with 5 decimal points of precision. Since most of the exchanges quote a price that is accurate up to 3 decimal points, the configuration and no price explicit definition ensure, that submitted prices fall near the perceived fair market price, while still leaving room for competition on the last decimals.

One of the unique features of the Flare Network is the ability for token holders to delegate their votes to data providers. This means that even if a token holder does not actively participate in the estimation process, they can still earn FTSO rewards by delegating their voting power [8] and impact the price by selecting a specific data provider. It is important to note, however, that the voting power of a single data provider is limited to 2.5% to avoid too big of an individual impact.

The FTSO's reward mechanism is fostering decentralization and ensuring real-time data accuracy. Given that the core task revolves around predicting prices of other providers, participants not only need to make accurate predictions but also strategize

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

to outperform others, making it a game of strategic decision-making. This challenge intriguingly sits at the crossroads of data science and game theory [6].

4 DATA RETRIEVAL AND PREDICTION

4.1 Overview

The data retrieval process is a crucial step in our analysis. It involves collecting, processing, and preparing time series data, specifically price and timestamp pairs, for further analysis. This data is essential for understanding trends, making predictions, and deriving insights.

The primary source of our data are the FTSO prices from previous epochs and current data from various exchanges. Selecting a specific subset of exchanges as a datasource is a nontrivial task. Each exchange has its own set of characteristics: trading volume, user base, regional influences, and even specific trading behaviors. Historical data shows, that providers are quick (on a sub-hour basis) to adapt to market opening and closing times and usually disregard after-hours trading prices on exchanges. Furthermore, the reliability of data from each exchange can vary. Some exchanges might offer more consistent and clean data, while others might have gaps or anomalies.

4.2 Data Processing and Smoothing Techniques

Once the data is retrieved, it undergoes several processing steps to ensure its quality and relevance for prediction. One of the primary challenges in time series forecasting is the inherent noise present in the data. Financial data is specifically prone to short-term spikes as low liquidity exchanges can experience large price deviations when market depth is limited. The spikes are quickly exploited by arbitragers, but price jumps - anomalies - are still available in the data and must be accounted for. We employ various smoothing techniques to filter out noise and highlight the underlying trends.

Exponential Moving Average (EMA): EMA is a type of weighted moving average that gives more weight to the most recent prices. In our system, the EMA vector and its alpha value are optimized using the `curve_fit` method from `scipy.optimize` library [10].

Savitzky-Golay Smoothing: This technique uses convolution to fit successive subsets of adjacent data points with a low-degree polynomial. It's effective in preserving the features of the distribution, such as heights and widths, making it suitable for our analysis [12].

Linear Interpolation: Linear interpolation is used to estimate values between two known values in a dataset. Our system employs a skew linear fit to interpolate missing or anomalous data points.

FFT Smoothing: The last smoothing method we've used is the Fast-Fourier smoothing.

Each of these methods has its own strengths and is chosen based on the specific characteristics of the data and the prediction requirements. So far, the only other smoothing method we've tried to incorporate is LOWESS (Locally Weighted Scatterplot Smoothing), which performed worse than the rest of the smoothing methods after training an overdetermined system on it (see 4.3). The mentioned methods were selected, as they are commonly used for smoothing the financial data [9], easily available in multiple scientific libraries, and offer good resilience against sudden spikes that are markets with low liquidity.

4.3 Prediction Mechanism

After smoothing the data using the techniques listed above, we adopt an overdetermined system approach for our predictions. This entails constructing a system of equations from the processed data and subsequently employing the least squares method to find the optimal prediction parameters.

Suppose we're training our time series over m epochs. Let $E \in \mathbb{R}^{m \times n}$ be a matrix where each column e_i , represents the price vector for the i-th exchange across the m epochs. Vector $\mathbf{v} \in \mathbb{R}^n$ signifies the normalized weights or contributions of each exchange to the forecasted price. Each entry, v_i in $\mathbf{v}$ corresponds to the significance of the i-th exchange.

Given the extensive epoch training data required for our model training and the limited availability of crypto exchanges (in the tens), we are dealing with an overdetermined system. In this context, we optimize the vector $\mathbf{v}$ using the least squares error method. The residual sum of squares evaluation function is optimized using the `fmin_cg` method from `scipy.optimize`, aiming to find the parameters that minimize the difference between the predicted values and the actual values in the training data.

For each exchange and for each smoothing method, we define a possible upper and lower range for the method's parameters and specify a step size. We then compute the cartesian product of all these sets, yielding all viable optimized parameter combinations in the form of a multidimensional grid. For each combination in this cartesian product, we smooth the data using the methods described above, train the model and calculate the optimal solution vector, which tells us how much weight should each exchange hold. Finally, we identify the model configuration that delivers the best performance.

The overdetermined system was chosen due to a number of different factors. We preferred a simple model with the potential for an explanation or at least the possibility of quick access to information in which input parameters offer greater prediction power. Although not included in our numerical utility function, delegation and the social aspect of goodness of price are important for multiple reasons. Being less good, but providing reasonable prices attracts more delegations and provides more security and trust in the network. Therefore, the error of not predicting the price fully correctly versus being off by a lot due to an edge condition or overfitting a specific input parameter was much preferred. Furthermore, incoming network upgrades might force the providers to buy or sell assets on the price revealed (and not on market price) and this means that a large deviation from the correct price would also be financially problematic.

Lastly, the providers work in *bursts*. Most of the information-rich exchange data comes in just before the end of the epoch (last few seconds), so a longer evaluation time might mean we miss some information or be too late for the submission. Our internal analysis shows, that submission must be calculated at least 8-5 seconds before the end of each epoch to be reliably accepted by the network validators. (network latency usually requires a submission of the price a few seconds before the end of the epoch).

5 RESULT ANALYSIS

We evaluated the performance of our trained models by comparing them against three simpler prediction methods: *Last Seen Value Method* predicts that the future value of a coin will be the most recent exchange price observed before the prediction starts. The *Previous Epoch Value* method predicts the price of a coin as

the FTSO price from the previous epoch. Lastly, we also try the overdetermined system witouh any smoothing.

Our calculation accuracy analysis spanned over a week, with new models trained every day on the previous 8-hour data (160 epochs). Following this, the model's success rate was then validated against the subsequent 8-hour dataset right after the training data. The success rate is the amount of times the predicted price would be in the interquartile range divided by the number of epochs the price was submitted for. This exactly corresponds to what price providers are financially incentivized to do.

The detailed results are presented inFigures 1a to 1d. As anticipated, the **Last Seen Value Method** method yields modest outcomes, averaging averaging prediction success rate of 3.5% across all coins.

For the **Previous Epoch Value Method** approach, we set the prediction to match the price from the previous epoch. While this method outperformed the first, it still registered a low performance, averaging around 7% for all coins over the week. Notably, several coins like *ETH* or *FIL* had an average success rate close to 0%, while *DOGE* achieved an average of 15%.

The method **Training an Overdetermined System Without Smoothing the Data** outperformed the first two, averaging around 10% success rate accross all coins during the testing week. Notably, the full prediction method that **Smooths the Data and Trains and Overdetermined System** outperformed all of the previous methods.

The evaluation closely mirrored real-world conditions, due to changes in exchanges, fluctuations in vote powers, and inclusion of new data providers in the median calculation, models must be continuously retrained on an almost daily basis. Over the observed epochs, our FTSO provider demonstrated varied success rates across different cryptocurrencies. The success rates for *XRP*, *DOGE* and *BTC* generally ranged between 0.20 to 0.45, indicating moderate to high prediction accuracy. Meanwhile, coins like *XLM*, *ADA*, and *ARB* had lower success rates, often below 0.15, suggesting challenges in predicting their prices. Overall, the provider's performance fluctuated across epochs and coins, with some cryptocurrencies consistently achieving higher success rates than others. Overall, we were able to achieve moderate prediction success of around 0.22, currently ranking 26th among the 94 active FTSO providers.

Because this method of smoothing and training an overdetermined system yielded better results than previous method of just training an overdetermined system, we can also be certain that smoothings in this case improve the result. This goes to show that without smoothing, our prediction model is highly influenced by noise and short-term fluctuations, making it challenging to capture the underlying trend in the time series data.

Coin	Last Seen	Prev. Ep	No smoth	Smooth
XRP	0.07412964	0.01536945	0.00542317	0.00398449
XLM	0.00010802	0.00025230	0.00090994	0.00025548
DOGE	0.00004626	0.00001359	0.00000733	0.00000641
ADA	0.00000201	0.00000395	0.00000183	0.00000174
BTC	23.78687273	5.01065648	1.94068887	0.91171693
ARB	0.00098386	0.00025156	0.00015229	0.00014042

Table 1: RMSE for each method and selected coins

Over the observed epochs, our FTSO provider demonstrated varied success rates across different cryptocurrencies. The success rates for *XRP*, *DOGE* and *BTC* generally ranged between 0.20 to 0.45, indicating moderate to high prediction accuracy. Meanwhile, coins like *XLM*, *ADA*, and *ARB* had lower success rates, often below 0.15, suggesting challenges in predicting their prices. Overall, the provider's performance fluctuated across epochs and coins, with some cryptocurrencies consistently achieving higher success rates than others. Overall, we were able to achieve moderate prediction success of around 0.22, currently ranking 26th among the 94 active FTSO providers.

Because this method of smoothing and training an overdetermined system yielded better results than previous method of just training an overdetermined system, we can also be certain that smoothings in this case improve the result. Without smoothing, our prediction model is highly influenced by noise and short-term fluctuations.

Coin	Last Seen	Prev. Ep	No smoth	Smooth
XRP	0.02129	0.04986	0.18729	0.339
XLM	0.02886	0.11686	0.03129	0.11329
DOGE	0.07686	0.16986	0.13186	0.38086
ADA	0.04143	0.14214	0.06157	0.13457
BTC	0.01043	0.01943	0.14071	0.32543
ARB	0.027	0.02343	0.09129	0.11529

Table 2: Average success rate for prediction methods

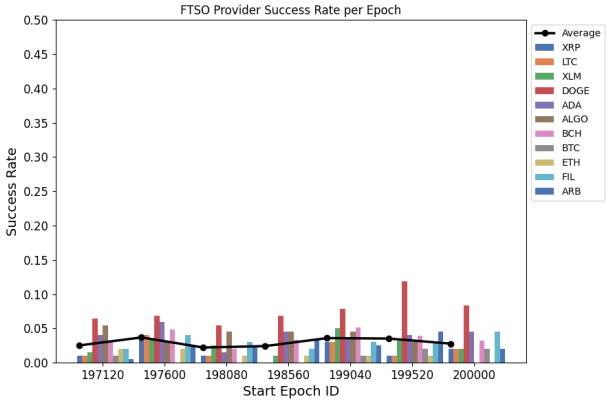
6 RMSE VALUES

Lastly, analyzed for each method and for each coin what is it's RSME (root mean squared error) to provide more insight into each method's accuracy. The results are depicted in 1. It's worth mentioning that since the prices of different coins vary, the RMSE values aren't comparable across the coins but only across the methods for one coin. For most coins, the *Last Seen Value* method generally yields the highest RMSE values, indicating the worst accuracy relative to other methods. Conversely, the *Overdetermined system with smoothing* method tends to produce the lowest RMSE values for most of the coins. The methods *Previous Epoch Value* and *Overdetermined system without smoothing* are ranked somewhere in between.

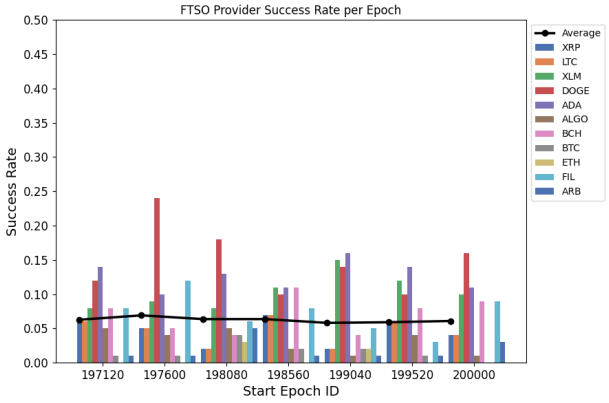
7 DISCUSSION AND FUTURE WORK

We have developed and assessed a functional provider solution to predict prices within the FTSO protocol. While we observed commendable performance for coins such as *XRP*, *DOGE*, and *BTC*, the results for other coins like *XLM*, *ADA*, and *ARB* were not as promising. Exploring additional smoothing techniques and incorporating multiple prediction methods would be beneficial. Notably, ensemble methods are renowned for reducing prediction variance, which in turn increases the probability of predictions falling within the median target range.

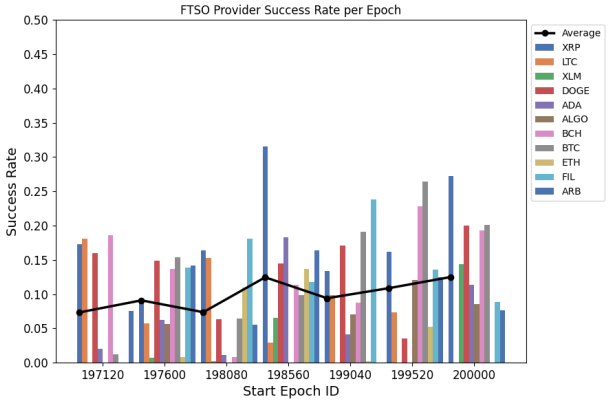
This paper has only focused on non-deep learning approaches to FTSO price prediction. A promising extension to the provider would be to explore time series prediction using various deep learning methods such as RNN or LSTM neural networks. These models have the potential to capture more subtle patterns in the data and adapt to the dynamic prices of the crypto coins. They might need to be modified to adapt to the specifics of the FTSO system and quick retraining times. Combining the more expensive inference of neural networks with presented overdetermined system together with error bounds on prediction results might also offer a more performant composite algorithm that would be



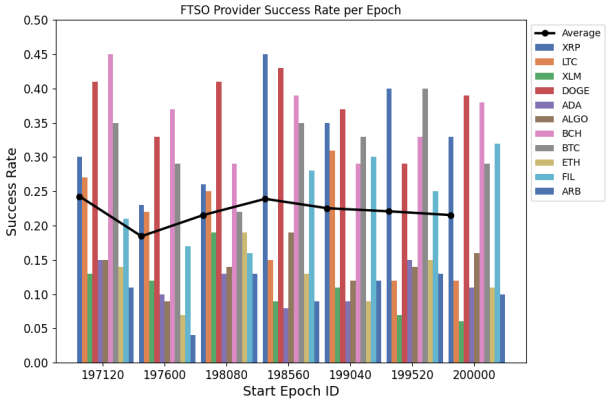
(a) "Last Seen Value" method



(b) "Previous Epoch Value" method



(c) Overdeterminded system without data smoothing



(d) Overdeterminded system without with data smoothing

able to use the fallback prediction in case of lateness of prediction by a stronger but more complicated model.

8 ACKNOWLEDGMENTS

The authors would like to thank AFLabs for the provision of exchange and FTSO data used during the development phase.

REFERENCES

[1] AlaghaA. "A reinforcement learning model for the reliability of blockchain oracles". In: *ScienceDirect* (2022).

[2] Giulio Caldarelli. "Overview of Blockchain Oracle Research". In: *MDPI* 14.6 (2022), p. 175.

[3] Giulio Caldarelli. "Understanding the Blockchain Oracle Problem: A Call for Action". In: *Information* 11.11 (2020), p. 509. URL: https://www.mdpi.com/2078-2489/11/11/509.

[4] Giulio Caldarelli. "Understanding the Blockchain Oracle Problem: A Call for Action". In: 11.11 (2023), p. 509.

[5] Chainlink. *Chainlink 2.0 and the future of Decentralized Oracle Networks*. Accessed: 2023-09-05. 2023. URL: https://chain.link/whitepaper.

[6] Vasant Dhar. "Data Science and Prediction". In: *Communications of the ACM* 56.12 (2013), pp. 64–73. URL: https://dl.acm.org/doi/abs/10.1145/2500499.

[7] Joshua Ellul. "The Blockchain Oracle Problem in Decentralized Finance—A Multivocal Approach". In: 11.16 (2023), p. 7572.

[8] Boi Faltings and Goran Radanovic. *Game Theory for Data Science: Eliciting Truthful Information*. Springer Nature, 2021.

[9] James D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994. URL: http://mayoral.iae-csic.org/timeseries2021/hamilton.pdf.

[10] A. J. Lawrance and P. A. W. Lewis. "An exponential moving-average sequence and point process (EMA1)". In: *Journal of Applied Probability* 14 (1 1977). Accessed: 2023-09-05, pp. 98–113. DOI: 10.2307/3213263.

[11] Christopher Potts. "Interpretive Economy". In: *Semantics Archive* (2008). Accessed: 2023-09-05. URL: https://semanticsarchive.net/Archive/jExYWZlN/potts-interpretive-economy-mar08.pdf.

[12] William H. Press and Saul A. Teukolsky. "Savitzky-Golay Smoothing Filters". In: *Computers in Physics and IEEE Computational Science & Engineering* 4.6 (1990). Accessed: 2023-09-05, pp. 669–672. DOI: 10.1063/1.4822961.

[13] Fan Zhang et al. "Decentralized Oracles: a Comprehensive Overview". In: *arXiv preprint arXiv:2004.07140* (2020). Accessed: 2023-09-05. URL: https://arxiv.org/abs/2004.07140.

[14] Yanchao Zhang Zhiqiang Liu and Jiwu Jing. "Connect API with Blockchain: A Survey on Blockchain Oracle Implementation". In: *ACM* (2022).

[15] Aviv Zohar and Ittay Eyal. "A Study of Blockchain Oracles". In: *arXiv* (2020). URL: https://arxiv.org/pdf/2004.07140.

On Neural Filter Selection for ON/OFF Classification of Home Appliances

Anže Pirnat and Carolina Fortuna
ap6928@student.uni-lj.si,carolina.fortuna@ijs.si
Jožef Stefan Institute, Ljubljana, Slovenia.

ABSTRACT

Non-intrusive load monitoring (NILM) enables the extraction of appliance-level consumption data from a single metering point. Appliance ON/OFF classification is a particular type of such appliance level data extraction recently enabled by deep learning (DL) techniques. To date, a study on the influence of neural filter selection on the performance and computational complexity for appliance ON/OFF classification is missing. In this paper, we start from a widely used DL architecture, adapt it for the appliance ON/OFF classification problem and then study the influence of the filters on the model performance and model complexity. Through this study we develop a model, PirnatCross, that excels at cross-dataset performance, offering an average improvement in average weighted F1 score of 17.2 percentage points vs a SotA model and VGG11 baseline respectively, when trained on REFIT and evaluated on UK-DALE and vice versa. Also, PirnatCross consumes 6-times less energy compared to a SotA model.

KEYWORDS

non-intrusive load monitoring (NILM), ON/OFF appliance classification, deep learning (DL), convolutional recurrent neural network (CRNN), multi-label classification

1 INTRODUCTION

Mitigating the impact of climate change is an urgent challenge that requires collective action to keep the global average temperature below 1.5°C in relation to pre-industrial levels. Reducing unnecessary electrical energy consumption and consequently limiting electrical energy production is a crucial step towards achieving our goals, as it is estimated that such activities account for over 40% of the total CO_2 equivalent generated by human activities [1]. Beside reducing energy consumption, we are increasingly adopting renewable power plants due to their significantly lower CO_2 emissions compared to fossil fuel-based ones [2]. However, renewable energy resources have a major drawback; dependency on renewable resources which are far less predictable, posing a challenge to the stability of the power system [11]. To address this issue, demand response strategies are being implemented to adjust electricity consumption to better match supply [1]. Consequently, efforts are being made to monitor and manage energy consumption more efficiently in residential buildings, making it relevant to track device activity (ON/OFF events) [3].

To avoid the high cost and invasiveness of monitoring each individual device with an electricity meter, researchers have developed a more economically efficient method known as non-intrusive load monitoring (NILM). This method involves obtaining appliance-level data using just one metering point to measure the total electricity consumption of a household. By using classification techniques for NILM, it is possible to determine the states (ON/OFF) of devices within a household and monitor their activity for demand response applications. As in a typical household it is possible to have several appliances working simultaneously, a suitable approach for determining the activity states of appliances is multi-label classification, where the state of each appliance is used as the class label and the recorded readings from a single household meter serve as input samples. Li *et al.* were among the first to propose multi-label classification for NILM disaggregation. More recently, Tanoni *et al.* [12] employed gated recurrent unit (GRU) in their CRNN for weakly-supervised training, mixing the amount of strongly and weakly labeled data to confirm the effectivness of such approach. Also Zhou *et al.* [14] proposed a new model called TTRNet, which uses a transpose convolution before a recurrent layer, a method, which has also shown better results in other works [8]. The existing works based on DL techniques typically lack a DL computational complexity/energy consumption analysis that is relevant in designing such models [2]. For instance, in [5] they analyzed the carbon footprint of various architectures and concluded that convolutional layers are power hungry because they operate in three dimensions, unlike fully connected layers which operate in two dimensions.

Existing studies typically develop and evaluate their method on a only a few datasets that are often limited in size. For instance [12] relied on two publicly available datasets and developed and evaluated a model for each of the two: REFIT [9] and UK-DALE [6]. While this approach is appropriate for relative method performance assessment, some studies have discussed also the importance of cross-dataset evaluation. For example, Han *et al.* [4] described significant dataset biases and high class imbalance of in-the-wild datasets as a fundamental bottleneck in facial expression recognition. Their results showed that cross-dataset evaluation can reduce dataset bias and improve the performance.

In this paper we aim to better understand the influence of the filters on the model performance and model complexity for multi-label ON/OFF appliance classification through intra and cross-dataset evaluation. Our main contributions are as follows:

- We adapt VGG19, a widely used DL architecture, for the appliance ON/OFF classification and study the influence of the filters on the model performance and model complexity.
- We develop a model, PirnatCross, that excels at cross-dataset performance, offering an average improvement of 17.2 percentage points vs a SotA model and VGG11 baseline respectively, when trained on REFIT and evaluated on UK-DALE and vice versa. Also, PirnatCross consumes 6-times less energy compared to SotA model.

[1] tinyurl.com/CO2-from-electricity1
[2] tinyurl.com/renewable-energy-doubled

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

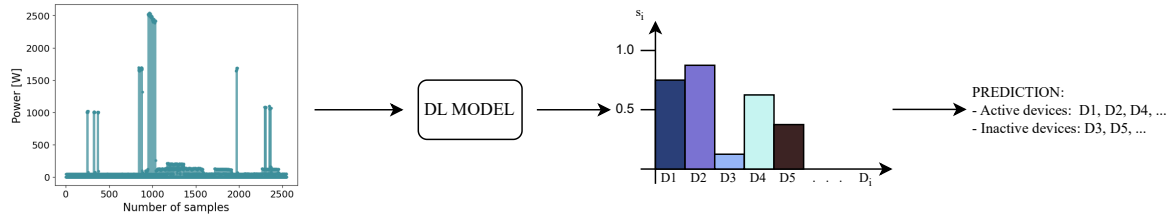


Figure 1: We input the data measured from a household into the DL model and it outputs s_i for each device present in the experiment. If s_i is greater than 0.5 we classify the device as active, if not as inactive.

The paper is organized as follows. Section 2 provides the problem statement, Section 3 presents methodological details, while Section 4 analyses the results of our study. Finally, Section 5 concludes the paper.

2 PROBLEM STATEMENT

Given an input power consumption measured by a smart meter $p(w)$ over a time window w, we aim to develop a multi-label ON/OFF classifier Φ that maps the input to a probability vector $s(w)$ corresponding to the status of the home appliances as:

$$s(w) = \Phi(p(w)) \tag{1}$$

The $|s|$ of the set s, indicates the number of appliances to be recognised. For each window of measurements $p(w)$ input to the model Φ, $s(w)$ will be of the form $[s_1(w), s_2(w), ..., s_N(w)]$, $s_i \in [0, 1]$ and $N = |s|$ where each s_i estimates the probability of appliance d_i to be active as also depicted in Figure 1. When $s_i > 0.5$ the appliance will be classified as ON, otherwise it will be classified as OFF. More than one appliance can be ON at the same time, therefor s contains multiple labels assigned to the current instance. In this paper $N = 5$ in total of which any 1-4 can be active.

The ON/OFF classifier Φ realized as a deep learning network is typically composed of a set of layers $[l_1, l_2, ...l_M]$ where the types of the layers may vary depending on how the respective architecture is designed. For instance $l_i \in [FC, Pool, Conv, GRU, ...]$, where FC stands for fully connected, Pool stands for pooling, Conv for convolutional and GRU for gated recurrent unit. As has been already shown also in [10], the computational complexity varies across the types of the layers.

In developing Φ, we start from the VGG family of architectures as they are widely used in various communities and have already shown promising results for classification on NILM [7]. More precisely we consider VGG19 comprising of 19 layers with trainable parameters, 16 of which are convolutional and 3 are fully connected. The convolutional layers are grouped into five blocks:

- Block 1: 2 x conv. with 64 filters + Max pooling
- Block 2: 2 x conv. with 128 filters + Max pooling
- Block 3: 4 x conv. with 256 filters + Max pooling
- Block 4: 4 x conv. with 512 filters + Max pooling
- Block 5: 4 x conv. with 512 filters + Max pooling

This architecture has been tailored to accommodate time series data, replacing the 2D convolutions and pooling from VGG19, designed for images, with 1D counterparts that are more suitable for time-series. In addition, the convolutional layers in the 5th block have been replaced with transpose convolutional layers to increase the temporal resolution of features to reduce their number as suggested in [14]. We also integrated a recurrent layer after the 5th block, GRU layer to be specific, as it is able to model temporal relationships in the time series and it was shown to achieve good performance in a recent study [12].

In order to estimate the computational complexity of the resulting architecture, referred to as PirnatCross, we must first calculate its complexity as the sum of all floating point operations (FLOPs) that have to be computed for each of its layers. This can be calculated for convolutional, pooling and fully-connected layers with the equations from [10] and for GRU with equation from [13].

As convolutional layers dominate in our adaptation of VGG19, and the computational complexity of a convolutional layer is relatively high compared to other type of layers [10]. Generally, the number of FLOPs used throughout the convolutional layer F_c is equal to the number of filters N_f times the flops per filter $F_c = (F_{pf} + N_{ipf})N_f$. Therefore we aim to study the influence of the number of the filters N_f on the model performance and complexity. Let the starting number of filters in each block of the adapted architecture be the same as in the original VGG19, namely $F = [64, 128, 256, 512, 512]$, analyze the model performance as average F1 score versus computational complexity in FLOPs.

3 METHODOLOGY

This section provides methodological details related to the datasets, the training approach and evaluation process that were employed for the study.

3.1 Datasets

The study is conducted using two datasets: UK-DALE [6] and REFIT [9]. Within each dataset, we monitor the same five appliances d_i that were also used in recent research [12]: fridge, washing machine, dishwasher, microwave, and kettle. The data from the selected devices is obtained and processed using the procedure described by Tanoni *et al.* [12] to form 2 mixed datasets. After processing, the two mixed datasets each consist of the same five devices, with each sample containing a random selection of one to four active devices. Samples with varying numbers of active devices are randomly distributed throughout the datasets. We evaluate the cross-dataset performance of models on two mixed datasets obtained by processing data from, UK-DALE and REFIT, in both directions. Specifically, we train models on REFIT derived dataset and test them on UK-DALE derived datasetand vice versa, by training on UK-DALE derived dataset and testing on REFIT derived dataset.

3.2 Benchmarks

In order to have a more meaningful study, we also evaluate PirnatCross, the adapted VGG19, against a VGG11 baseline and a recently published work TanoniCRNN [12]. For VGG11, we used a learning rate of 0.0001 and the same batch size and epochs. For

TanoniCRNN, we used the hyperparameters specified as optimal in their paper [12].

For PirnatCross we vary the set of the filters F by multiplying with $k \in [0.02, 0.04, 0.06, 0.08, 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5]$. The learning rate, batch size, and number of epochs were determined through a process of trial and error, informed by previous experiments, and subsequently fine-tuned for each model, to optimize model performance and stability. The resulting values are: learning rate of 0.0003, a batch size of 128, and trained for 20 epochs.

While some models were capable of handling larger batch sizes, we found that performance was not improved by increasing the batch size beyond 128, so we kept it unchanged for all models. We train and evaluate using 5-fold cross-validation.

3.3 Metrics

We use the average weighted F1 score ($\overline{F1score_w}$) as a performance metric because our datasets are not balanced and do not provide equal representation for each device.

$$\overline{F1score_w} = \sum_{i=1}^{N_d} F1score_i \times Weight_i \quad (2)$$

The average weighted F1 score is calculated using three metrics: true positive (TP), false positive (FP), and false negative (FN). TP measures the instances where the device is accurately classified as active, while FP represents cases where the device is erroneously classified as active. FN indicates instances where the device is mistakenly classified as inactive.

From these metrics, we derive the precision ($Precision = \frac{TP}{TP+FP}$) and recall ($Recall = \frac{TP}{TP+FN}$), which are used to calculate the F1 score ($F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$). To obtain the average weighted F1 score (2), we first compute the F1 score for each device, then take the average based on their weight ($Weight = \frac{SSD}{SAD}$), which is determined by the support for the specified device (SSD) and the support of all devices (SAD).

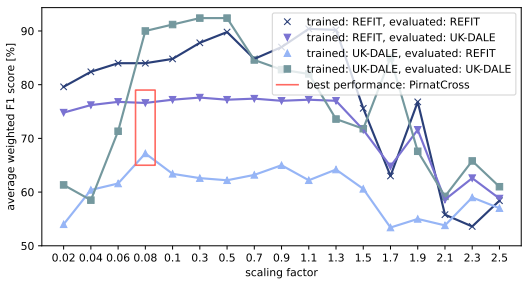


Figure 2: Average F1 scores on intra and cross-dataset training and evaluation as a function of filter scaling factor.

4 RESULTS

In this section we first determine the optimal filter configuration for variations of the PirnatCross architecture to achieve high average weighted F1 score. We then follow with a computational complexity and carbon footprint assessment. Finally, we then benchmark the performance of models in cross-dataset evaluation on REFIT and UK-DALE datasets.

4.1 Analysis of Tuning the Filters in PirnatCross

Figure 2 depicts the performance of the PirnatCross architecture where the original number of filters in the set F has been scaled by scaling factors $k \in [0.02, 0.04, ..., 2.5]$. The upper two curves present the average weighted F1 score for models trained and evaluated on REFIT and UK-DALE separately, so without cross-dataset evaluation. The second lowest curve presents the average weighted F1 scores for models trained on REFIT and cross evaluated on UK-DALE while the lowest curve presents the results on training on UK-DALE and cross evaluating on REFIT. In our experiments, we observe only the cross evaluation models, they show a rapid improvement in performance for scaling factor values from 0.02 to 0.08. From scaling factor value 0.08 to 0.9, we see a decline in performance in one example and a small improvement in the others, while beyond 0.9 the results gradually decline. For scaling factors above 1.3 a rapid drop in performance can be observed.

Marked with light blue in Figure 2 and also depicted in Figure 3 is the PirnatCross version of the proposed architecture having F scaled by 0,08 and thus resulting in the $F_1 = [5, 10, 20, 40, 40]$ filter configuration of the blocks. PirnatCross1 performs optimally in terms of avg F1 score.

PirnatCross1 also contain 5 blocks as the original VGG19. The first two comprising of two convolutional layers and the subsequent two comprising of four convolutional layers. The final block consists of four transpose convolutional layers and all blocks include an average pooling layer after the convolutional layers. Preceding the output layer, our model incorporates a GRU layer with a size of 64. Additionally, two fully-connected layers, each consisting of 4096 nodes, are included in the architecture. The output layer of our model comprises five nodes corresponding to the states s_i of the 5 appliances d_i considered in this study. All layers utilize the ReLU activation function, except for the output layer which employs the sigmoid activation function.

4.2 Computational Complexity and Carbon Footprint Analysis

Table 1 summarizes the weights, FLOPs, energy and carbon footprint numbers for PirnatCross versus the TanoniCRNN and VGG11 baselines. The results take into account the fact that the models were trained on Nvidia A100 graphics card, located in Slovenia where 250g of CO_2 equivalent is produced with each kWh of electricity. The specific equations used to calculate, energy and carbon footprint are defined in our previous work [10].

It can be seen from the second row of the table that PirnatCross achieves superior energy efficiency compared to other models, exhibiting energy consumption 6-times smaller compared to SotA TanoniCRNN and 6.6-times smaller compared to VGG11.

Table 1: Computational complexity and carbon footprint analysis for the proposed architecture and selected baselines.

NN	weights	FLOPs	energy	carbon footprint
PirnatCross	$17.4 \cdot 10^6$	$\mathbf{185 \cdot 10^6}$	**329 kJ**	**22,9 g CO_2 eq.**
TanoniCRNN [12]	$\mathbf{0.75 \cdot 10^6}$	$1.11 \cdot 10^9$	1967 kJ	136.7 g CO_2 eq.
VGG11	$185.6 \cdot 10^6$	$1.21 \cdot 10^9$	2150 kJ	149.3 g CO_2 eq.

4.3 Cross-Dataset Analysis

Tables 2 and 3 present the per device breakdown of the F1 scores for PirnatCross, TanoniCRNN and VGG11 when trained on REFIT and evaluated on UK-DALE and vice versa.

When we trained on REFIT and evaluated on UK-DALE, the scores for the four models were as follows: PirnatCross achieved a score of 0.766, TanoniCRNN achieved a score of 0.752 and VGG11

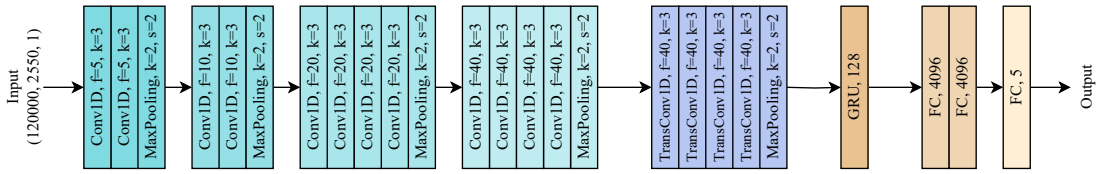


Figure 3: The proposed architecture PirnatCross made for maximum performance.

Table 2: F1 scores for PirnatCross1, TanoniCRNN [12] and VGG11 trained on REFIT and evaluated on UK-DALE.

devices	PirnatCross	TanoniCRNN [12]	VGG11
fridge	0,944	0,972	0,462
washing machine	0,650	0,690	0,544
dish washer	0,646	0,648	0,294
microwave	0,728	0,756	0,512
kettle	0,786	0,622	0,420
weighted avg	**0,766**	0,752	0,456

Table 3: F1 scores for PirnatCross1, TanoniCRNN [12] and VGG11 trained on UK-DALE and evaluated on REFIT.

devices	PirnatCross	TanoniCRNN [12]	VGG11
fridge	0,730	0,232	0,508
washing machine	0,668	0,666	0,366
dish washer	0,596	0,468	0,360
microwave	0,526	0,630	0,506
kettle	0,800	0,782	0,408
weighted avg	**0,672**	0,542	0,438

achieved a score of 0.456. However, when we trained on UK-DALE and tested on REFIT, the scores were notably lower for all four models. PirnatCross achieved a score of 0.672, TanoniCRNN achieved a score of 0.542, and VGG11 achieved a score of 0.438.

This outcome may be explained by the fact that REFIT has a significantly higher level of data noise compared to UK-DALE as shown in prior work [12]. Consequently, the testing results obtained from UK-DALE are expected to show higher F1 scores. Moreover, we observed that, overall, our model PirnatCross consistently outperformed the other models in both testing scenarios, achieving the highest weighted average F1 scores overall.

5 CONCLUSIONS

To address the challenge of cross-dataset usage scenario on NILM ON/OFF classification, we propose PirnatCross, with an aim to present the maximum performance and the energy efficiency. The results of our evaluation on the REFIT and UKDALE datasets reveal that PirnatCross achieve an average performance improvement of 7.2 over SotA and 27.2 percentage points over baseline, underscoring its superior effectiveness in handling data from diverse sources. Additionally PirnatCross consumes 6-times less energy compared to SotA model. To develop PirnatCross, we employed our methodology. In the case of classification on NILM this included beginning with the VGG19 architecture and implementing several modifications, such as replacing the convolutional layers with transpose convolutional layers in the 5th block, incorporating a GRU layer after it, and adjusting the number of filters based on our analysis. Our analysis revealed that an increase in the number of filters in convolutional layers and consequently an increase in the number of FLOPs did not necessarily lead to an improvement in classification accuracy. Instead, we observed a point of steady improvement in performance, followed by a gradual decline and a significant drop in performance when the number of filters exceeded a certain threshold. This information is crucial for optimizing the architecture of NILM models, and keeping track of the carbon footprint.

ACKNOWLEDGEMENTS

This work was funded in part by the Slovenian Research Agency under the grant P2-0016. The authors would like to thank Blaž Bertalanič for insightful discussions.

REFERENCES

[1] Jamshid Aghaei and Mohammad-Iman Alizadeh. 2013. Demand response in smart electricity grids equipped with renewable energy sources: a review. *Renewable and Sustainable Energy Reviews*, 18, 64–72. DOI: https://doi.org/10.1016/j.rser.2012.09.019.

[2] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. 2019. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134, 75–88. DOI: https://doi.org/10.1016/j.jpdc.2019.07.007.

[3] R. Gopinath, Mukesh Kumar, C. Prakash Chandra Joshua, and Kota Srinivas. 2020. Energy management using non-intrusive load monitoring techniques – state-of-the-art and future research directions. *Sustainable Cities and Society*, 62, 102411. DOI: https://doi.org/10.1016/j.scs.2020.102411.

[4] Byungok Han, Woo-Han Yun, Jang-Hee Yoo, and Won Hwa Kim. 2020. Toward unbiased facial expression recognition in the wild via cross-dataset adaptation. *IEEE Access*, 8, 159172–159181.

[5] Gigi Hsueh. 2020. *Carbon Footprint of Machine Learning Algorithms*. Senior Projects Spring 2020. 296. https://digitalcommons.bard.edu/senproj_s2020/296.

[6] Jack Kelly and William Knottenbelt. 2015. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2, 1, 1–14.

[7] Weicong Kong, Zhao Yang Dong, Bo Wang, Junhua Zhao, and Jie Huang. 2020. A practical solution for non-intrusive type ii load monitoring based on deep learning and post-processing. *IEEE Transactions on Smart Grid*, 11, 1, 148–160. DOI: 10.1109/TSG.2019.2918330.

[8] Luca Massidda, Marino Marrocu, and Simone Manca. 2020. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. *Applied Sciences*, 10, 4. DOI: 10.3390/app10041454.

[9] David Murray, Lina Stankovic, and Vladimir Stankovic. 2017. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Scientific data*, 4, 1, 1–12. DOI: 10.1038/sdata.2016.122.

[10] Anže Pirnat, Blaž Bertalanič, Gregor Cerar, Mihael Mohorčič, Marko Meža, and Carolina Fortuna. 2022. Towards sustainable deep learning for wireless fingerprinting localization. In *ICC 2022 - IEEE International Conference on Communications*, 3208–3213. DOI: 10.1109/ICC45855.2022.9838464.

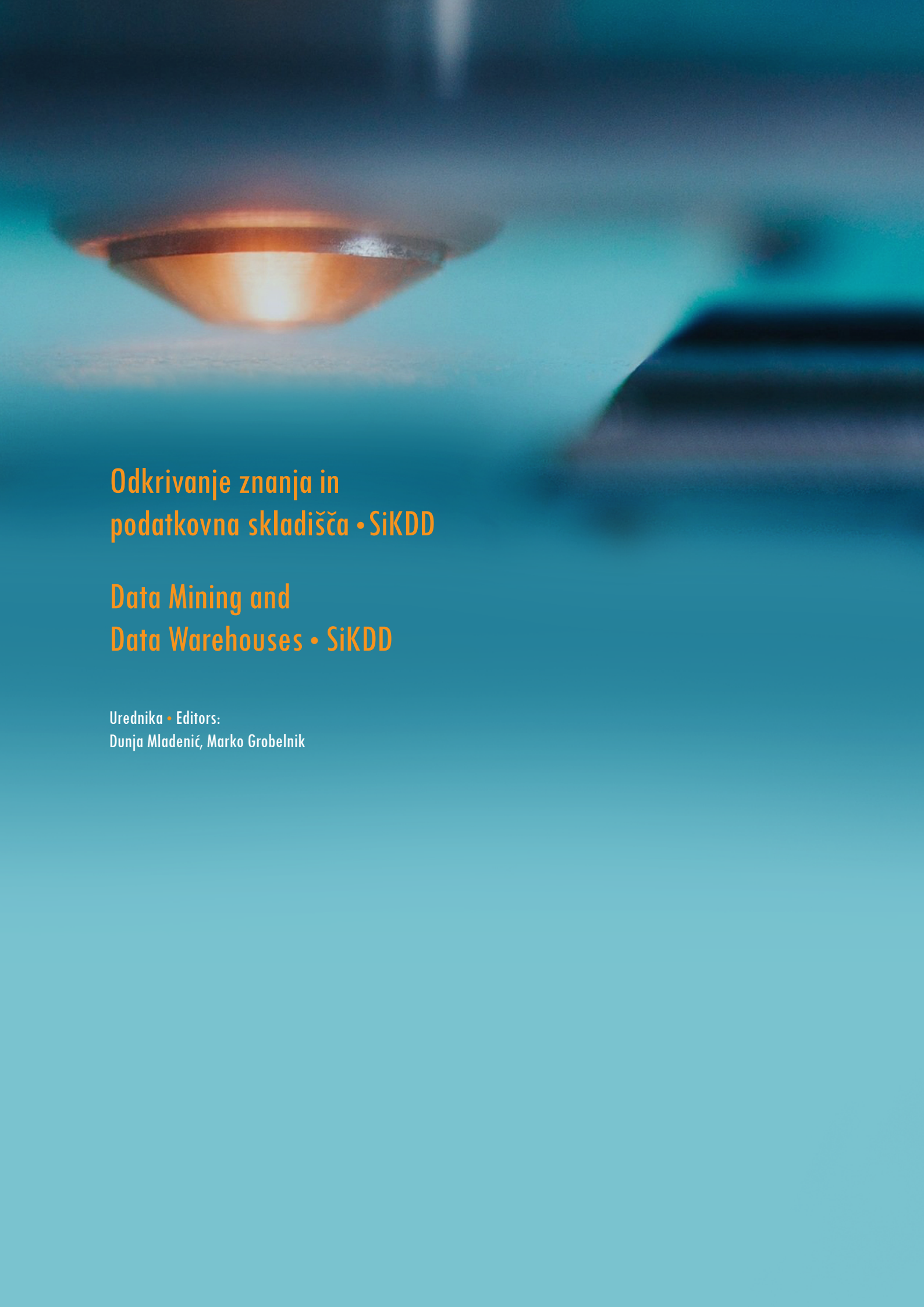
[11] Ali Q. Al-Shetwi, M.A. Hannan, Ker Pin Jern, M. Mansur, and T.M.I. Mahlia. 2020. Grid-connected renewable energy sources: review of the recent integration requirements and control methods. *Journal of Cleaner Production*, 253, 119831. DOI: https://doi.org/10.1016/j.jclepro.2019.119831.

[12] Giulia Tanoni, Emanuele Principi, and Stefano Squartini. 2022. Multi-label appliance classification with weakly labeled data for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 1–1. DOI: 10.1109/TSG.2022.3191908.

[13] Minjia Zhang, Wenhan Wang, Xiaodong Liu, Jianfeng Gao, and Yuxiong He. 2018. Navigating with graph representations for fast and scalable decoding of neural language models. *Advances in neural information processing systems*, 31.

[14] Mengran Zhou, Shuai Shao, Xu Wang, Ziwei Zhu, and Feng Hu. 2022. Deep learning-based non-intrusive commercial load monitoring. *Sensors*, 22, 14. DOI: 10.3390/s22145250.

Indeks avtorjev / Author index



Odkrivanje znanja in podatkovna skladišča • SiKDD

Data Mining and Data Warehouses • SiKDD

Urednika • Editors:
Dunja Mladenić, Marko Grobelnik