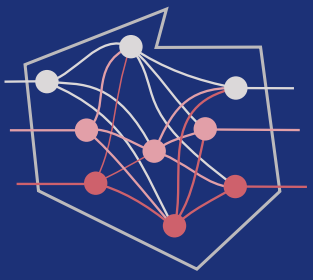
PROGRESS IN POLISH ARTIFICIAL INTELLIGENCE RESEARCH 5

Edited by:
Jacek Mańdziuk
Adam Żychowski
Mikołaj Małkiński

Proceedings of the 5th Polish Conference on Artificial Intelligence (PP-RAI'2024)

18–20.04.2024, Warsaw, Poland

PP-RAI'2024



Warsaw University of Technology
Faculty of Mathematics and Information Science

Progress in Polish Artificial Intelligence Research 5

Edited by:
Jacek Mańdziuk, Adam Żychowski, Mikołaj Małkiński

Warsaw 2024

Cover design by
Andrzej Kowalczyk

Publisher: Warsaw University of Technology
WUT Press (Oficyna Wydawnicza Politechniki Warszawskiej, UIW 48800),
Polna 50, 00-644 Warsaw, phone. (48) 22 234-70-83

ISBN 978-83-8156-696-4 (print)
ISBN 978-83-8156-697-1 (online)
DOI: 10.17388/WUT.2024.0002.MiNI

Order No. 249/24
Printing and bookbinding in WUT Press Printers, phone (48) 22 234 70 30

Preface

This book contains the proceedings of the 5th Polish Conference on Artificial Intelligence (PP-RAI'2024) which was held at the Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland, on April 18-20, 2024.

The conference was organized by the Warsaw University of Technology, in cooperation with the Polish Alliance for the Development of Artificial Intelligence (*pol.: Polskie Porozumienie na rzecz Rozwoju Sztucznej Inteligencji* – PP-RAI), under the patronage of the Rector of Warsaw University of Technology.

PP-RAI was established in 2018 as an initiative of the Polish scientific community engaged in research in the field of artificial intelligence (AI) and machine learning (ML). The main goal of PP-RAI is integration of the Polish AI/ML community and facilitation of the development of AI/ML in Poland. The alliance includes five member organizations:

- Polish Artificial Intelligence Society
- Polish Neural Network Society
- Polish Special Interest Group on Machine Learning
- Polish Chapter of the IEEE Systems, Man, and Cybernetics Society
- Polish Chapter of the IEEE Computational Intelligence Society

Annual PP-RAI conferences aim to bring together researchers from the area of AI/ML and provide a platform for:

- sharing, exchange, presentation, and discussion of original research results in different areas of AI/ML;
- discussion on important research initiatives, projects, and events related to AI/ML development in Poland and abroad;
- cooperation between the AI/ML community, research institutions and business partners.

The 2024 edition of the PP-RAI conference featured 3 plenary lectures delivered by the top researchers in the field, 2 panels devoted to contemporary topics in AI, 4 poster sessions and 3 oral parallel sessions with 188 presented papers altogether, divided into 14 thematic tracks. These proceedings include 75 out of 188 accepted articles, designated by the Track Chairs, based on the content and research potential.

I would like to thank all the Authors of submitted papers for their efforts in paper preparation and presentation, the Reviewers for their dedicated service for the community, the Track Chairs and the Program Committee for the final paper selection, and the Steering Committee for their guidance throughout the whole process. Last but not least, I am grateful to the Local Organizing Committee for their constant support in local organization of the event. Finally, I would like to acknowledge the support from our sponsors, and above all from the Excellence Initiative: Research University (IDUB) program at the Warsaw University of Technology.

Jacek Mańdziuk

PP-RAI'2024 Conference Chair

April, 2024

Organization

GENERAL CHAIR

Krzysztof Zaremba	Rector of Warsaw University of Technology

STEERING COMMITTEE

Ireneusz Czarnowski	Gdynia Maritime University
Włodzisław Duch	Nicolaus Copernicus University in Toruń
Janusz Kacprzyk	Systems Research Institute of the Polish Academy of Sciences, Warsaw
Jacek Mańdziuk	Warsaw University of Technology
Grzegorz J. Nalepa	Jagiellonian University, Krakow
Leszek Rutkowski	Systems Research Institute of the Polish Academy of Sciences, Warsaw
Rafał Scherer	Czestochowa University of Technology
Jerzy Stefanowski	Poznan University of Technology
Adam Wojciechowski	Lodz University of Technology
Michał Woźniak	Wrocław University of Science and Technology
Beata Zielosko	University of Silesia in Katowice

PROGRAM CHAIR

Jacek Mańdziuk	Warsaw University of Technology

PROGRAM COMMITTEE

Jarosław Arabas	Warsaw University of Technology
Aleksander Byrski	AGH University of Krakow
Leszek Chmielewski	Warsaw University of Life Sciences
Ireneusz Czarnowski	Gdynia Maritime University
Włodzisław Duch	Nicolaus Copernicus University in Toruń
Krzysztof Gajowniczek	Warsaw University of Life Sciences
Tomasz Gambin	Warsaw University of Technology
Maria Ganzha	Warsaw University of Technology
Maciej Grzenda	Warsaw University of Technology
Agnieszka Jastrzębska	Warsaw University of Technology
Janusz Kacprzyk	Systems Research Institute of the Polish Academy of Sciences
Stanisław Kaźmierczak	Warsaw University of Technology

Piotr A. Kowalski	AGH University of Krakow
Dariusz Krol	Wrocław University of Science and Technology
Marcin Kurdziel	AGH University of Krakow
Halina Kwaśnicka	Wrocław University of Science and Technology
Bogdan Kwolek	AGH University of Krakow
Piotr Lipiński	Lodz University of Technology
Agnieszka Ławrynowicz	Poznan University of Technology
Szymon Łukasik	AGH University of Krakow
Mikołaj Małkiński	Warsaw University of Technology
Jacek Mańdziuk	Warsaw University of Technology
Grzegorz J. Nalepa	Jagiellonian University
Robert Nowak	Warsaw University of Technology
Karol Opara	Systems Research Institute of the Polish Academy of Sciences
Piotr Pęzik	University of Lodz
Maciej Piasecki	Wrocław University of Science and Technology
Dariusz Plewczyński	Warsaw University of Technology
Jacek Rumiński	Gdańsk University of Technology
Leszek Rutkowski	Systems Research Institute of the Polish Academy of Sciences
Khalid Saeed	Bialystok University of Technology
Rafał Scherer	Czestochowa University of Technology
Piotr Skrzypczyński	Poznan University of Technology
Przemysław Spurek	Jagiellonian University
Jerzy Stefanowski	Poznan University of Technology
Dominik Ślęzak	University of Warsaw
Julian Szymański	Gdańsk University of Technology
Tomasz Trzciński	Warsaw University of Technology
Piotr Wasilewski	Systems Research Institute of the Polish Academy of Sciences
Jarosław Wąs	AGH University of Krakow
Adam Wojciechowski	Lodz University of Technology
Michał Woźniak	Wrocław University of Science and Technology
Cezary Zieliński	Warsaw University of Technology
Beata Zielosko	University of Silesia in Katowice
Maciej Zięba	Wrocław University of Science and Technology
Adam Żychowski	Warsaw University of Technology

LOCAL ORGANIZING COMMITTEE
(Local Team - Warsaw University of Technology)

Maria Ganzha
Stanisław Kaźmierczak
Mikołaj Małkiński
Jacek Mańdziuk - Chair
Cezary Zieliński
Maciej Żelaszczyk
Adam Żychowski

Reviewers:

Adrian Weronika
Arnaiz-González Álvar
Artiemjew Piotr
Balicki Jerzy
Batorski Paweł
Belter Dominik
Bentkowska Urszula
Blachnik Marcin
Bobek Szymon
Borycki Piotr
Byrski Aleksander
Cejrowski Tymoteusz
Charytanowicz Małgorzata
Chmielewski Leszek
Cygert Sebastian
Czarnowski Ireneusz
Dobrosolski Jan
Dolata Przemysław
Domański Paweł
Doroz Rafał
Drygaś Paweł
Dubiński Jan
Duch Włodzisław
Dudek Wojciech
Ernst Sebastian
Ewald Dawid
Ganzha Maria
Hachaj Tomasz
Hernes Marcin
Jobczyk Krystian
Jodłowiec Marcin
Jóźwiak Rafał
Kaleta Joanna
Kania Adam
Kania Kacper
Karanowski Konrad
Karwowski Jan
Kasprzak Włodzimierz
Kasymov Artur
Kaźmierczak Stanisław
Kepski Michal
Kicki Piotr
Klikowski Jakub
Klukowski Piotr
Kluza Krzysztof
Komorniczak Joanna
Kornuta Tomasz
Kozakiewicz Adam
Kozłowski Wojciech
Kraft Marek
Krol Dariusz
Książek Kamil
Ksieniewicz Paweł
Kudłacik Przemysław
Kurdziel Marcin
Kwolek Bogdan
Lazarowska Agnieszka
Lesiński Wojciech
Lipinski Piotr
Łaźniewski Michał
Łukasik Szymon
Majcher Mateusz
Majek Karol
Majkutewicz Jan
Malec Jacek
Maleszka Marcin
Matusiewicz Zofia
Migacz Dawid
Mikołajczyk-Bareła Agnieszka
Morawski Michał
Mroczek Teresa
Nalepa Grzegorz
Napieralski Piotr
Niewiadomska-Szynkiewicz Ewa
Nowak Przemyslaw
Nowak Robert
Nowak-Brzezinska Agnieszka
Nowicki Michał
Ogrodniczuk Maciej
Okarma Krzysztof

Olewniczak Szymon
Opara Karol
Orczyk Tomasz
Paja Wiesław
Paprzycki Marcin
Pawlak Stanisław
Pawłowski Wiesław
Pekala Barbara
Pełka Paweł
Pezik Piotr
Piasecki Maciej
Pikus Michał
Podsędkowski Leszek
Potoniec Jedrzej
Rygiel Patryk
Rypeść Grzegorz
Sendera Marcin
Seredyński Dawid
Siemiatkowska Barbara
Sienkewicz Elzbieta
Siwek Krzysztof
Skinderowicz Rafał
Skruch Paweł
Skrzypczynski Piotr
Slot Krzysztof
Stefanowski Jerzy
Stefańczyk Maciej
Stokfiszewski Kamil
Stypułkowski Michał
Szarmach Marta
Szklarski Jacek
Szpyrka Marcin
Szymanski Julian
Ślażyński Mateusz
Śluzek Andrzej
Świechowski Maciej
Świrta Bartosz
Tadeja Slawomir
Tobis Sławomir
Tomaszewska Paulina
Tomczyk Arkadiusz
Waczyńska Joanna
Walczak Jakub
Wasilewski Piotr
Wąs Jarosław
Wielopolski Patryk
Wilkowski Artur
Wisniewski Dawid
Wojciechowski Adam
Wojciechowski Szymon
Wojnicki Igor
Wozniak Marcin
Wozniak Michal
Woźniak Piotr
Wróblewska Anna
Wysoczańska Monika
Zachara Marek
Zamorski Maciej
Zielinska Teresa
Zielinski Cezary
Zielonka Adam
Zyblewski Paweł
Żelaszczyk Maciej
Żychowski Adam

Contents

Chapter 1

Data Mining and Machine Learning

Track Chairs:

- prof. Jerzy Stefanowski - Poznan University of Technology
- prof. Michał Woźniak - Wrocław University of Science and Technology
- prof. Ireneusz Czarnowski - Gdynia Maritime University

A-PETE: Adaptive Prototype Explanations of Tree Ensembles

Jacek Karolczak[0000−0001−5414−960X], **Jerzy Stefanowski**[0000−0002−4949−8271]

Poznan University of Technology
ul. Piotrowo 3, 60-965 Poznan, Poland
jacek.karolczak@student.put.poznan.pl jstefanowski@cs.put.poznan.pl

Abstract. *The need for interpreting machine learning models is addressed through prototype explanations within the context of tree ensembles. An algorithm named Adaptive Prototype Explanations of Tree Ensembles (A-PETE) is proposed to automatise the selection of prototypes for these classifiers. Its unique characteristics is using a specialised distance measure and a modified k-medoid approach. Experiments demonstrated its competitive predictive accuracy with respect to earlier explanation algorithms. It also provides a a sufficient number of prototypes for the purpose of interpreting the random forest classifier.*
Keywords: *machine learning, explainable AI, prototypes, random forest*

1. Introduction

In recent years, there has been a growing need to explain machine learning black box models, which has led to the development of the field of explainable Artificial Intelligence (XAI) [1]. Making machine learning models' decisions interpretable, enables users to trust and rely on models' predictions. In general, one can distinguish between *global* and *local explanations*. The global explanations refer to studying more comprehensive interactions between input attributes and output, target variables for the complete data set, rather aimed at a general explanation of how ML model works. On the other hand, the local explanation attempts to discover the reasons for a single decision of the model for a given instance. [1].

While explanations can be given in different forms, this study focuses on explanations articulated via *prototypes*. As defined in [2] prototypes are certain instances (representative real examples or synthetic ones) defined on the basis of the

DOI:10.17388/WUT.2024.0002.MiNI

learning dataset to explain the behaviour of machine learning models, or sometimes also to explain the underlying data distribution. A user can comprehend the model's reasoning by comparing prototypes to user's data. Having a well identified set of prototypes, an application-specific explanation can be yielded more easily than more complex explanations. For instance, prototypes can act as a global interpretation of the underlying behaviour of an obscure black box ML model, by presenting to the user the limited number of representative prototypes for a class. Prototypes can also be used for local explanation of a single decision, by showing the similarity of the predicted instance to the nearest or max few nearest representative exemplar(s) – prototype(s) [3]. Local explanation can be further enhanced by showing the most similar prototypes belonging to other classes, to give deeper insight into the decision.

Inspired by Tan et al [4], we will consider the new idea of finding such prototypes for explaining Random Forest predictions, which uses a specialised distance measure and a modified k-medoids algorithm. However, it is important to identify a limited number of such prototypes due to the perceptual abilities of the recipient, which was not always preserved in experiments [4]. Furthermore, we believe that the user should be supported in this task. Therefore, the main goal of our paper includes studying the automatic selection of the number of these prototypes. The usefulness of this algorithm (named A-PETE) and earlier proposals are experimentally evaluated.

2. The distance function for tree ensembles

Our work extends the proposal of a similarity measure for instances within a tree ensemble [4], which considers predictions and distributions of learning examples in the tree leaves. The proposed approach works under the condition of equal contribution of each tree to the final decision. Let t represent the number of trees in the tree ensemble (TE). The i-th tree ($i \in [t]$) partitions the feature space into regions $R_{i,j}$, each corresponding to a leaf $\tau_{i,j}$. Each tree induces an individual classifier assigning each point $x \in X$ to a single region $R_{i,j}$:

$$c_i^{\text{Tree}}(x) = \sum_{j=1}^{\tau_i} \alpha_{i,j} \mathbb{1}(x \in R_{i,j}), \tag{1}$$

where $\alpha_{i,j}$ is the predicted value in the j-th leaf of the i-th tree. $\mathbb{1}$ denotes the indicator function. The tree ensemble classifier is the average over all trees:

$$c^{\mathrm{TE}}(x) = \frac{1}{t}\sum_{i=1}^{t} c_i^{\mathrm{Tree}} = \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{\tau_i} \alpha_{i,j}\mathbb{1}(x \in R_{i,j})\,. \tag{2}$$

Thus, the proximity of two instances x_1 and x_2 is given as the mean number of trees in which both instances land in the same leaf and can be expressed as

$$\mathrm{proximity}^{\mathrm{TE}}(x_1, x_2) = \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{\tau_i} \mathbb{1}(x_1 \in R_{i,j})\mathbb{1}(x_2 \in R_{i,j})\,. \tag{3}$$

The distance metric can be derived from the proximity function $d^{\mathrm{TE}}(x_1, x_2) = 1 - \mathrm{proximity}^{\mathrm{TE}}(x_1, x_2)$.

Given that the regions $\{R_{i,j}\}_{j=1}^{\tau_i}$ form a partition of the feature space, each point $x \in X$ belongs to at most one region. The inner sum in the proximity definition yields values of either 0 or 1 for each tree. Thus, the proximity, a convex combination of all trees, and consequently the distance, also lies within the range of [0, 1].

3. Prototype selection as medoids

The set of discovered prototypes P should satisfy the following properties:

- Reality – P consists of real objects from the learning set ($P \subset X$).
- Coverage – For a class c, the prototypes in P^c should collectively span the class, ensuring that each instance $x^c \in X^c$ is represented by (close to) a prototype in P^c ($P^c = \arg\min_{p^c \in X^c} \sum_{i=1}^{|P^c|}\sum_{j=1}^{|X^c|} \mathrm{distance}^{\mathrm{TE}}(p_i^c, x_j^c)$).
- Compactness – The set of prototypes should be the minimal collection of instances that satisfies the aforementioned conditions ($|P| \ll |X|$).

This work is inspired by the work of Tan et al [4], where the K-medoids problem was adapted to the task of prototype selection, that is the objective is to find a subset of examples $P \subset X$, where $|P| = k$. However, a drawback of [4] is that the user has to define the number of medoids (prototypes) k.

We propose the Adaptive Prototype Explanations of Tree Ensembles (A-PETE), which automatically selects k prototypes. We adopt the greedy submodular prototype selection algorithm (SM-A), which minimises the function:

$$f(P) = \sum_{c}^{C}\sum_{i}^{|X^c|} \min_{p^c \in P^c} d(x_i^c, p^c)\,. \tag{4}$$

Algorithm 1: Adaptive Prototype Explanations of Tree Ensembles (A-PETE).

Input : Set of points X, distance function $d : X^2 \mapsto [0, 1]$, class assignment $c : X \mapsto [q]$, control parameter $\alpha \in (0, 1)$
Output: Set of prototypes P

Create set of phantom exemplars $P' = \{p'_1, ..., p'_q\}$ and set $d(p'_i, x) = d(x, p'_i) = 1$ for all $x \in X$
$\Delta \leftarrow 0$
$\mathrm{P} \leftarrow \varnothing$
while *True* **do**
  $x^* \leftarrow \arg\max_{x \in X}[f(P') - f(P' \cup P \cup \{x\})]$
  $\Delta' \leftarrow f(P' \cup P \cup \{x^*\}) - f(P' \cup P)$
  $P \leftarrow P \cup \{x^*\}$
  if $\frac{|\Delta - \Delta'|}{\Delta'} < \alpha$ **then**
    break
  end
  $\Delta' \leftarrow \Delta$
end

The A-PETE algorithm is composed of three parts. Initially, a set of phantom prototypes P' is created, where the distance between these prototypes and all $x \in X$ equals 1. Additionally, an empty set is prepared for real prototypes P. The core of the algorithm lies in the iterative prototype selection process. In each iteration, the algorithm identifies an instance x^* that maximises the difference between the total distance of phantom prototypes P' and the set is a union of phantom prototypes P', prototypes P and the maximising instance x^*. The maximising instance x^* is added to the set of prototypes P. The algorithm computes the change in the objective function Δ resulting from adding the selected instance x to the set of prototypes.

The main novelty is that our algorithm maintains the difference Δ between consecutive objective function changes to monitor the progress. If the relative change surpasses the threshold, the algorithm concludes, and the set of prototypes P is the final output. A comprehensive pseudocode is presented as Algorithm 1.

Table 1. The values of p (the number of features in nodes of Random Forest) yielding best results (Table 2), found in hyperparameter tuning.

	Breastcancer	Diabetes	Compass	RHC	Mnist	Caltech256
SM-A	sqrt	0.33	0.33	7	sqrt	sqrt
SM-WA	sqrt	sqrt	0.5	0.5	sqrt	0.33
A-PETE	sqrt	0.33	0.5	7	0.33	sqrt

4. Experimental evaluation

We will compare our proposal A-PETE against the earlier two proposals of selecting prototypes (SM-A and SM-WA) coming from [4].

The SM-A, SM-WA, and A-PETE were implemented on top of the Random Forest ensemble with scikit-learn Python library[1]. To be consistent with hyperparameters from [4], the random forests were trained with 1000 trees without limiting the tree depth. The number of features was tuned from values: $\sqrt{p}$, 7, $0.33p$, $0.5p$, $0.7p$, where p is the number of features. The values of p yielding the best results, found during hyperparameter tuning, for each algorithm is presented in Table 1.

The SM-A and SM-WA algorithms were executed with a search scope of up to 20 prototypes. This choice was guided by our empirical study revealing that a higher number of prototypes may pose challenges for interpretation due to inherent limitations in cognitive capabilities. However, it should be noted that SM-A and SM-WA algorithms could produce a much higher number of final prototypes (see the original results published in [4]).

We chose 6 popular datasets, often used in the related studies (in particular also in [4]). These data have different sizes and characteristics. Due to pages limits we only gave their names in Table 2 and the reader y can find more details about them. Note that for CALTECH-256, two easily confused classes guitar and mandolin were selected and deep features were extracted by ResNet-50 trained on ImageNet. For Mnist digits 4 and 9 as look-alike classes were selected and raw pixel values were used as features. Compass contains several discriminative features, so it was preprocessed following the strategy proposed in AIF-360[2]. All categorical features of RHC and Compass underwent an ordinal encoding.

[1] https://scikit-learn.org

[2] https://github.com/Trusted-AI/AIF360/

Table 2. The best weighted accuracy achieved using Random Forest (RF) and 1-NN run on prototypes selected using adaptive greedy submodular prototype selection (SM-A), weighted adaptive greedy submodular prototype selection (SM-WA), and Adaptive Prototype Explanations of Tree Ensembles (A-PETE). The number of prototypes in parentheses.

	Breastcancer	Diabetes	Compass	RHC	Mnist	Caltech256
RF	0.93	0.73	0.66	0.75	0.99	0.69
SM-A	0.92 (8)	0.74 (3)	0.30 (20)	0.74 (12)	0.97 (14)	0.70 (16)
SM-WA	0.92 (8)	0.72 (2)	0.30 (20)	0.40 (10)	0.97 (11)	0.72 (5)
A-PETE	0.92 (7)	0.73 (5)	0.32 (23)	0.73 (9)	0.97 (19)	0.70 (6)

Having no real user case study, we decided to use a surrogate model approach to assessing the predictive ability, i.e. the fidelity measure in relation to the Random Forest prediction. For all the prototype selection algorithms, the test example is assigned to the class of the nearest prototype. Such an assessment procedure is consistent with Tan et al. [4]. For Mnist the original train-valid-test split was maintained. For the remaining datasets, the train-valid-test split was done with proportions 60%-20%-20% and class stratification.

The A-PETE approach was compared to the adaptive greedy submodular (SM-A) and weighted adaptive greedy submodular (SM-WA) introduced in [4] using weighted – balanced accuracy as the evaluation metric suitable to deal with class imbalances [5]. It is computed as a weighted sum of the accuracy of each class, where the weight of each class is determined by dividing the number of instances of that class by the total number of instances in the dataset. It is noteworthy that A-PETE was executed for Breastcancer, Diabetes, Mnist, and Caltech256, utilising the default $\alpha = 0.05$. Due to sub-optimal outcomes for Compass and RHC, A-PETE was subject to additional evaluation with $\alpha = 0.01$.

5. Results

The experiment highlighted the effectiveness of automatically selecting prototypes. In terms of weighted (balanced) accuracy, the surrogate (nearest prototype) model achieved almost the same predictive performance for both SM-A and A-PETE on half of the datasets. For the Diabetes and RHC datasets, A-PETE provided a weighted accuracy only one percentage point lower than SM-A. In contrast,

A-PETE outperformed SM-A by two percentage points in the case of the Compass dataset, indicating the negligibility of the difference. Except for Compass, all three algorithms achieved accuracy levels comparable to the original Random Forest ensemble, which suggests that prototypes may capture essential information for imitating the Random Forest's decision-making process.

Moreover, A-PETE consistently approached the number of prototypes from SM-A without sacrificing predictive performance, suggesting sufficient prototype selection. In contrast, SM-A still requires manual tuning of the optimal k value.

To sum up, A-PETE offers a promising solution for the automatic prototype selection in tree ensembles, addressing the need for the local and global explanation of the tree ensembles.

Ack: The research on this paper has been partially supported by PUT SBAD grant.

References

[1] Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., and Rinzivillo, S. Benchmarking and Survey of Explanation Methods for Black Box Models. *Data Mining and Knowledge Discovery*, pages 1–60, 2023.

[2] Molnar, C. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[3] Obermair, C., Fuchs, A., Pernkopf, F., Felsberger, L., Apollonio, A., and Wollmann, D. Example or prototype? learning concept-based explanations in time-series. In E. Khan and M. Gonen, editors, *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, pages 816–831. PMLR, 2023.

[4] Tan, S., Soloviev, M., Hooker, G., and Wells, M. T. Tree space prototypes: Another look at making tree ensembles interpretable. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, FODS '20, page 23–34. Association for Computing Machinery, 2020.

[5] Brzezinski, D., Stefanowski, J., Susmaga, R., and Szczech, I. Visual-based analysis of classification measures and their properties for class imbalanced problems. *Information Sciences*, 462:242–261, 2018.

Model Discovery with Grammatical Evolution. An Experiment with Prime Numbers

Jakub Skrzyński[1][0009−0008−4550−5009], **Dominik Sepioło**[1][0000−0001−7746−3781], **Antoni Ligęza**[1][0000−0002−6573−4246]

[1]*AGH University of Krakow*
Department of Applied Computer Science
al. A. Mickiewicza 30, 30-059 Krakow, Poland
jskrzynski@student.agh.edu.pl, {sepiolo, ligeza}@agh.edu.pl

Abstract. *Machine Learning produces efficient decision and prediction models based on input-output data only. Such models have the form of decision trees or neural nets and are far from transparent analytical models, based on mathematical formulas. Analytical model discovery requires additional knowledge and may be performed with Grammatical Evolution. Such models are transparent, concise, and have readable components and structure. This paper reports on a non-trivial experiment with generating such models.*
Keywords: *machine learning, model discovery, grammatical evolution, explainable artificial intelligence, model-driven XAI*

1. Introduction

The need for development of transparent, white-box, explainable models in Artificial Intelligence becomes more and more visible [1]. In Machine Learning (ML), the classical decision trees induction and neural nets learning supplemented with the shallow methods of post-hoc eXplainable Artificial Intelligence (XAI) [2] might be replaced by Model-Driven approaches [3], [4]. Model-Discovery seems to be a promising direction in Artificial Intelligence in domains where understanding *how-it-works* is required [5, 6].

One of the methods enabling development of transparent models is *Grammatical Evolution* (GE) [7, 8]. GE is a kind of genetic algorithm that can perform *Symbolic Regression* tasks and find the best combination of operations to discover a formula that describes a given phenomenon as accurately as possible.

DOI:10.17388/WUT.2024.0002.MiNI

GE is an evolutionary algorithm that utilizes additional knowledge, provided as a Context-Free Grammar (CFG), to perform genotype-phenotype mapping, allowing limiting the search space and simultaneously increasing the search efficiency. CFG design has a direct impact on search efficiency and correctness of given solutions, but at the same time, it gives huge flexibility in terms of describing the problem. Depending on CFG, the result could be a mathematical expression, architecture of a neural network, or a syntactically correct computer program. GE is a candidate approach for achieving explainability better than some shallow approaches [4].

The main aim of this paper is to demonstrate the use of GE in search of white-box model based on existing partial knowledge of problem structure, expressed as CFG, and relatively small sets of collected data. For the purpose of the experiment, the function known for lack of exact formula was chosen to emphasize the approximation capabilities of GE.

The approach outlined in this paper can be applied in real-life scenarios where estimation of an unknown function is needed with only partial knowledge available. This method is particularly useful for identifying relationships between variables in complex phenomena that are difficult to analyze analytically. Examples include the discovery of relation between medical parameters, or describing phenomena observed in physics and much more.

2. Experiment

Function $\pi(x)$ is defined as the amount of *prime numbers* less or equal to x [9]. Despite its significance, the precise calculation of the $\pi(x)$ function remains a challenging endeavor due to the nature of prime numbers and their distribution. The elusive and irregular pattern of prime numbers presents a formidable computational hurdle, leading to the absence of an exact analytical expression for $\pi(x)$.

Grammatical Evolution emerges as a promising method for evolving mathematical formulas that endeavor to capture the intricate nature of prime number distribution. Unlike traditional approaches, Grammatical Evolution harnesses the power of evolutionary algorithms to iteratively refine and optimize mathematical expressions, seeking a formula that best approximates $\pi(x)$.

A series of experiments, conducted using the Python programming language leveraging the PonyGE2 library [10], aimed to construct models that seek to unveil the $\pi(x)$ function, based on example data and mean squared error (MSE) as the

fitness function, is reported.

2.1. The Grammar

To evolve formulae approximating $\pi(x)$, a grammar describing mathematical expressions is needed. For demonstration purposes, we utilized the grammar provided by PonyGE2 with slight modifications. These adjustments, made to align with the specific requirements of our experiment, are presented below:

```
<e>   ::=  <e>+<e>|<e>-<e>|<e>*<e>|
           pdiv(<e>,<e>)|
           psqrt(<e>)|
           np.sin(<e>)|
           np.tanh(<e>)|
           np.exp(<e>)|
           plog(<e>)|
           x[:, 0]|
           <c><c>.<c><c>
<c>   ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

2.2. The Data Set

In order to generate an example dataset, a C++ program was implemented. The program produces a hard-coded array of prime numbers which can be referred to during operation to optimize time consumption. Prime numbers may be taken from internet sources. Generated data set quality depends on the number of prime numbers. Data generated by the program should be stored in a text file, in a format required by PonyGE2, that will be later referenced in the parameters file.

To prepare the dataset, a subset of prime numbers was selected, including primes within the range of $< 2; 7919 >$, resulting in a total of 1000 entries.

3. The Results

Throughout the experiment, multiple functions resembling the shape of $\pi(x)$ were evolved. However, a challenge arises in the complexity of the evolved formulas, characterized by numerous nested operations. An example of one such result is presented below:

```
2 sqrt(x) + x/(tanh((x + sqrt(tanh(78.45) sin(51.98))
    x - log(sqrt(84.76) + 47.5))/exp(log(log(69.92) +
    7.51)) x) + sqrt(38.86) + log(log(x - log(sin(x) +
    15.6) tanh(tanh(sqrt(x))) tanh(sin(log(x)) + 69.37)
    x)))
```

The complexity issue can be addressed, for example, by imposing stricter constraints on the depth of the derivation tree. However, this approach may affect the accuracy of the results.

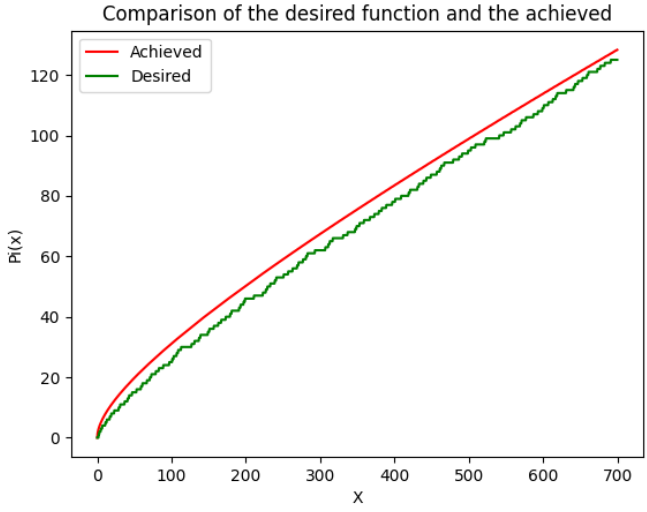


Figure 1. First solution generated with GE compared to actual function.

As it can be seen in Fig. 1, the generated solution does have a similar shape to the desired function. However, the approximation is not perfect, showcasing differences between predicted values and actual ones. Subsequent executions of the experiment, with a higher number of generations, were conducted, yielding the following results:

```
x/(ln(x/(ln(ln(92.89 - sin(x)+x*x+sin(x) -64.03* sqrt(x)*
    ln(exp(sin(89.77))))*sqrt(sin(19.94))))))
```

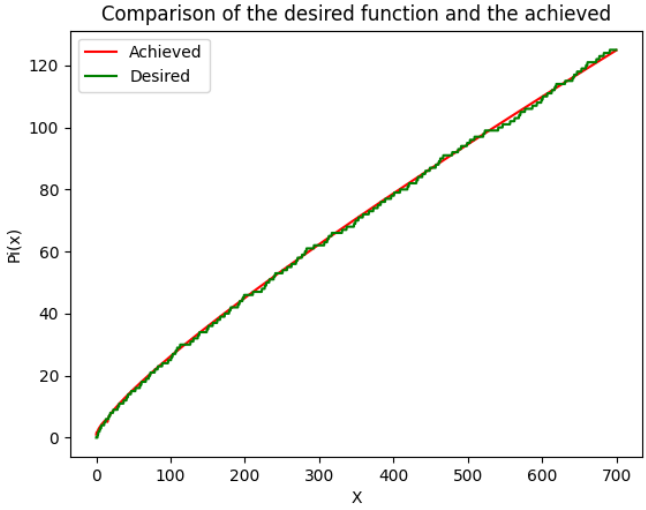


Figure 2. Second generated solution in comparison to actual function.

While the results of the simple experiment are reasonably satisfying, differences persist. Table 1 compares function values for two x values, highlighting that the evolved function is close but does not precisely describe the function. To gain a visual overview, one may refer to Fig. 2.

x	$f_2(x)$	$\pi(x)$
100	26.0574	25
1400	222.801	222

Table 1. Values of $\pi(x)$ in comparison to values of second evolved function

The results of the experiment show greater promise in terms of execution time, with the evolution of the second formula taking only 143.7 seconds. Considering the achieved effectiveness relative to resource consumption, there is a potential to enhance precision by expanding the search space. This could lead to an even more refined approximation of the given function. Furthermore, supplying a larger dataset might help maintain the function shape across a broader range of values.

4. Further work

Considering the successful application of performing regression in order to find the approximate form of an unknown function of one parameter, further research may focus on applications of this technique to more complex scenarios. This includes examining relationships involving multiple independent variables and one or more dependent variables, with a constant focus on enhancing accuracy.

5. Conclusions

The performed experiments show just one of many potential use cases, but emphasize the advantages of GE and also potential issues that may emerge during the development of the GE model. Grammars (and their definitions) are candidate tool to introduce additional knowledge into the system. Careful design may limit the search space in accordance with already possessed knowledge about the system.

Designing a correct model requires deep knowledge of the utility of the solution. It allows the user to create a correct fitness function that is being used to rate evolved solutions, therefore allowing the selection of the correct results.

GE models allow injecting knowledge into the system and have the efficiency of retrieving good solutions proportional to the amount of introduced knowledge. GE does not require a lot of theory knowledge from the final users, enabling them to adopt the technique relatively quickly.

References

[1] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019. doi:10.1038/S42256-019-0048-X.

[2] Arrieta, A. B. et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2019.

[3] Sepioło, D. and Ligęza, A. Towards model-driven explainable artificial intelligence. an experiment with shallow methods versus grammatical evolution.

In *Artificial Intelligence. ECAI 2023 International Workshops*, pages 360–365. Springer Nature Switzerland, Cham, 2024.

[4] Sepioło, D. and Ligęza, A. A comparison of shallow explainable artificial intelligence methods against grammatical evolution approach. In *Progress in Polish Artificial Intelligence Research 4*, pages 89–94. Lodz University of Technology Press, 2023.

[5] Ligęza, A., Jemiolo, P., Adrian, W. T., Slazynski, M., Adrian, M., Jobczyk, K., Kluza, K., Stachura-Terlecka, B., and Wisniewski, P. Explainable artificial intelligence. model discovery with constraint programming. In M. Stettinger and et al, editors, *Intelligent Systems in Industrial Applications, 25th International Symposium, ISMIS 2020, Graz, Austria*, volume 949 of *Studies in Computational Intelligence*, pages 171–191. Springer, 2020.

[6] Ligęza, A. and Sepioło, D. In search for model-driven explainable artificial intelligence. In *ECAI 2023 International Workshops: Invited presentation at the AI for Sustainability*. To be published, Krakow, 2024.

[7] O'Neill, M. and Ryan, C. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358, 2001.

[8] Ryan, C., O'Neill, M., and Collins, J. J., editors. *Handbook of Grammatical Evolution*. Springer, 2018.

[9] Shanks, D. *Solved and unsolved problems in number theory*, volume 297. American Mathematical Soc., 2001.

[10] Fenton, M., McDermott, J., Fagan, D., Forstenlechner, S., Hemberg, E., and O'Neill, M. Ponyge2: grammatical evolution in python. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, page 1194–1201. Association for Computing Machinery, 2017.

On Feature Importance in Tree-Based Machine Learning Models

Witali Bułatow[0000−0002−3635−1094], **Marek Krętowski**[0000−0001−9175−2678]

Bialystok University of Technology
Faculty of Computer Science
Wiejska 45A, 15-351 Bialystok, Poland
witali.bulatow@pb.edu.pl
m.kretowski@pb.edu.pl

Abstract. *This paper proposes a novel approach called Example-Based Importance to quantify feature importance scores for decision trees using instances from the training set. The method can be adapted to ensemble tree-based classifiers, such as Random Forest. Additionally, three new techniques are introduced to determine feature importance for an arbitrary regressor trained on datasets containing numerical features. The proposed methods are evaluated on publically available regression datasets and compared with well-known algorithms for feature importance calculation, Mean Decrease Gini and Mean Decrease Accuracy.*
Keywords: *feature importance, decision tree, regression*

1. Introduction

The concept of feature importance has received significant attention in the field of explainable machine learning, which aims to develop models that can be easily understood and interpreted. The importance of interpretable machine learning [1] has been acknowledged by the research community, and there is a growing interest in developing methods that can provide insights into the decision-making process of complex models.

In the field of feature importance, several notable scientific achievements have been made over the years. One of the most influential results was Breiman's 2001 paper on Random Forests [2], which introduced the concept of variable importance measures based on the mean decrease in impurity (MDI) criterion. Since then,

DOI:10.17388/WUT.2024.0002.MiNI

various methods have been proposed to address potential biases and limitations of the MDI criterion, such as the conditional variable importance measure proposed by Strobl et al. in [3].

In 2007, Strobl et al. in [4] also identified potential bias in variable importance measures when using Random Forests, which led to the development of a bias-corrected measure. In [5], a method for explaining the predictions of machine learning models based on partial dependence plots was introduced. In addition, Gregorutti et al. proposed in [6] a correlation-based variable importance measure that takes into account the correlation between predictor variables.

This paper introduces novel approaches for calculating feature importance scores using observations from the training dataset that was used to build the model. The proposed methods are compared against the established *Mean Decrease Gini* (MDG) and *Mean Decrease Accuracy* (MDA) techniques [2].

2. Methods

In this section, we outline the methods employed in our study to assess feature importance using training instances in machine learning models. This approach is based on the fact that during the learning stage, the model interacts solely with training objects. Hence, it is a logical strategy to extract feature relevance from the training dataset. We present the *Example-Based Importance* (EBI) method, which quantifies feature importance in tree-based models. Additionally, we introduce three novel methods that are applicable to arbitrary regressors.

2.1. EBI

Consider a univariate binary decision tree, denoted as T, for either regression or classification on a training set with p features. At each node i, we define the feature being tested as f_i, and the number of training instances that reach this node as n_i. The nodes of the tree are arranged in a left-to-right and top-to-bottom order. It is worth noting that n_1 represents the total number of instances in the dataset. Our objective is to determine the importance score I_i for a given feature f_i, where i ranges from 1 to p.

To compute the importance score, we define the importance of each feature as follows. For each observation x_i in the training set, where i ranges from 1 to n_1, we examine the associated decision path $path_j$ in the tree T, where j ranges from 1 to s, the number of leaves in the tree. The length of each path $path_j$ is denoted as L_j.

For each feature f_k present in the decision path, we increment its raw importance score $\tilde{I}_k$ by $1/L_j$.

It is worth noting that a feature may appear multiple times in the decision path, which increases its relevance for decision-making. Therefore, we assume that features in the path have an equal contribution to the result, regardless of their hierarchy in the tree. Furthermore, all splitting variables in the decision path have a cumulative impact of 1 in the decision process. This ensures higher feature importance scores are assigned to the features that determine the split of a larger number of training objects. Additionally, features that are repeatedly used in the decision-making process are given higher importance scores.

To determine the importance of a specific feature f_i, we sum up the impacts obtained from each observation across the entire training dataset. The resulting importance scores are then normalized so that they add up to 1.

Alternatively, we can calculate the raw feature importance scores $\tilde{I}_i$ using the following formula, where ℓ_k represents the number of training instances that reached the k-th leaf:

$$\tilde{I}_i = \sum_{k=1}^{s} \left[\frac{\ell_k}{L_k} \cdot \sum_{j=1}^{L_k} [path_k(j) = f_i] \right].$$

Here, $[path_k(j) = f_i]$ is equal to 1 if the j-th term of the decision path sequence $path_k$ is equal to the feature f_i, and 0 otherwise. After computing the raw importance scores $\tilde{I}_i$ for each feature, we normalize them such that they add up to 1, using the following formula:

$$I_i = \frac{\tilde{I}_i}{\sum_{j=1}^{p} \tilde{I}_j}.$$

This normalization step ensures that different algorithms used to calculate feature importance scores are comparable.

2.2. Feature Importance for Regression Models

Now we introduce a method for computing feature importance scores for an arbitrary regressor, which has been trained on a standardized dataset containing only numerical variables. Our method involves taking a single observation from the training set and perturbing the value of a single observation at a given feature f_k and observing the effect of this perturbation on the outcome. By examining the

magnitude of the changes in prediction, we can propose a measure to quantify the importance score of the feature f_k.

Let C be an arbitrary regressor that has been trained on a standardized dataset, and let x_i be a particular observation in the training set. Our objective is to compute the importance score for a numerical feature f_k for this observation. We denote the value of an i-th observation x_i at the k-th predictor as $x_i(k)$ and the prediction of the regressor for this observation as $C(x_i)$. We modify the value of feature f_k by generating a grid of N perturbed observations x_{ik}^j, where j ranges from $-R$ to R with a step of $2R/(N-1)$. Here N and R are user-defined parameters with default values of $N = 21$ and $R = 1.0$. Each perturbed observation takes on at feature f_k equally spaced values $x_{ik}^j(k)$ within a closed interval of radius R centered at the original value $x_{ik}^0(k)$.

Next, we calculate the prediction of the regressor for each perturbed observation x_{ik}^j in a grid, resulting in a vector of length N that contains predictions $C\left(x_{ik}^j\right)$ for each perturbed observation in a grid. We repeat this process for all numerical features f_k, producing a grid of perturbed regression predictions of size $N \times n_{num}$, where n_{num} is the number of numerical features in the dataset.

We present three distinct approaches for measuring the importance of a numerical feature f_k based on observation x_i:

1. *Linear Regression* (LR): We fit a linear regression line to the perturbed observations, where on the x-axis we place the points $x_{ik}^j(k)$, and the corresponding regression predictions $C\left(x_{ik}^j\right)$ on the y-axis. We define the importance of feature f_k as the absolute value of the slope of the regression line.

2. *Sorted Linear Regression* (LRSORT): Similar to the first approach, we fit a linear regression line with the same x-axis coordinates. However, the y-axis coordinates are sorted in ascending order. The importance of feature f_k is defined as the slope of the regression line. This approach might be useful when the perturbed regression scores are not linear, therefore without sorting precedure we may not be able to capture the significance of potentially important feature.

3. *Differences in Predictions* (DIFF): We compute the absolute difference between consecutive perturbed regression scores in the grid and sum them. We define the importance of feature f_k as this sum of $N-1$ values.

Finally, we compute the raw feature importances $\tilde{I}_k$ using one of the methods described above for all n_1 training instances by adding all the partial results. Then the raw importances are normalized in the same way as before by dividing each of them by their sum.

3. Experiments

Experimental studies were conducted using the datasets whose characteristics are presented in Table 1.

Table 1. Characteristics of the selected datasheets

Dataset	Instances	Features
Diabetes	442	9
California	20640	8
Insurance	1338	2

The MDG algorithm measures the total reduction of impurity based on the Gini index when a given feature is used for a split. The MDA scores measure the mean decrease in model accuracy when a given feature is permuted. All the results have been normalized such that the sum of feature importances is equal to 1.

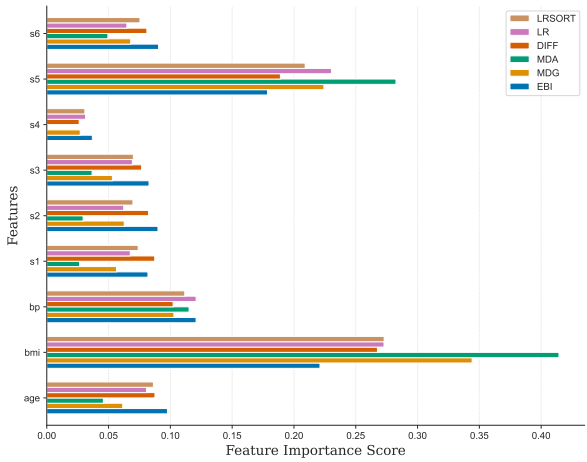


Figure 1. Comparison of the new methods with MDG and MDA algorithms on the *Diabetes* dataset

The results of the comparison between our proposed approach and the MDG and MDA algorithms for the *Diabetes* dataset are presented in Figure 1. It can be observed that all the methods exhibit comparable performance in identifying the most important numerical features. However, there are some subtle differences in their output. Specifically, MDG and MDA tend to yield larger differences in the importance scores between features. In contrast, our proposed methods achieve a more balanced distribution of importance scores, while still preserving the same order of the features. Overall, the differences between the algorithms are not significant, indicating that our proposed approach is a viable alternative to the MDG and MDA methods.

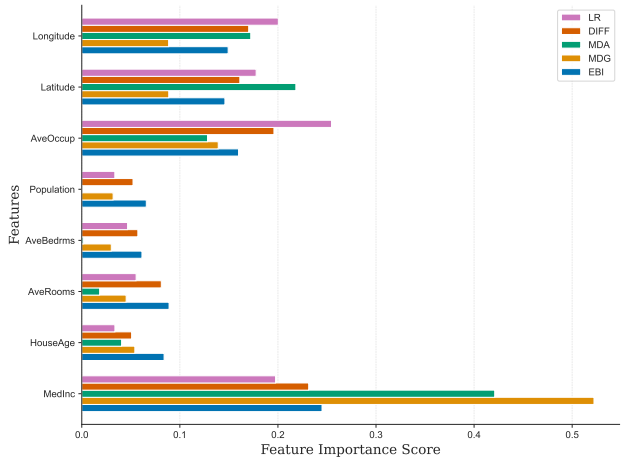


Figure 2. Comparison of EBI, MDG, MDA, DIFF, and LR algorithms for the *California Housing* dataset

Figure 2 displays the feature importance scores for five algorithms – EBI, MDG, MDA, DIFF, and LR – applied to the *California Housing* dataset using the Random Forest Regressor. The new algorithms suggest a slightly different feature as the most important one in comparison to the MDG and MDA methods. Namely, `AveOccup` is identified as the feature of the highest significance by the

new methods, whereas MDG and MDA methods rank `MedInc` as the most important feature.

Despite these differences in the rankings, it is worth noting that the feature importance scores obtained by all five algorithms are comparable. Therefore, the slight variation in the feature rankings may be due to the specific strengths and weaknesses of each algorithm, as well as the unique characteristics of the dataset.

4. Conclusions

In this article, we delve into the algorithms designed for calculating feature importance, which play a crucial role in identifying and understanding the significant features in a machine learning model. Nonetheless, it is challenging to compare the performance of different feature importance algorithms, primarily due to the lack of an appropriate metric that can comprehensively evaluate their effectiveness. The algorithms discussed in this article provide a useful starting point for researchers and practitioners to explore different methods for feature importance calculation and select the most appropriate one for their specific dataset and problem.

References

[1] Molnar, C. *Interpretable Machine Learning*. Leanpub, 2020.

[2] Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.

[3] Strobl, C., Boulesteix, A., Kneib, T., Augustin, T., and Zeileis, A. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307, 2008.

[4] Strobl, C., Boulesteix, A., Zeileis, A., and Hothorn, T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007.

[5] Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014. doi:10.1007/s10115-013-0679-x.

[6] Gregorutti, B., Michel, B., and Saint-Pierre, P. Correlation and variable importance in random forests. *Statistics and Computing*, 27:659–678, 2017.

Accelerating Machine Learning for Earth Observation Using Intel Extension for Scikit-learn

Bogdan Ruszczak[1,2][0000−0003−1089−1778],
Agata M. Wijata[2,3][0000−0001−6180−9979],
Jakub Nalepa[2,3][0000−0002−4026−1569]

[1]*Opole University of Technology, Opole, Poland*
b.ruszczak@po.edu.pl
[2]*KP Labs, Gliwice, Poland*
[3]*Silesian University of Technology, Gliwice, Poland*
{awijata,jnalepa}@ieee.org

Abstract. *Providing a means of rapid training and inference of machine learning models is of paramount importance in the world of Big Data. Accelerating classic models is equally important to deep learning ones, as they are deployed in a variety of tasks, e.g., on edge devices. We investigate the recent Intel Extension for a popular Scikit-learn package, promising to significantly speed up training and inference. We evaluated classification models over a dataset from the Earth observation domain, targeting bare soil detection. The results revealed that the models implemented using Intel's Extension run significantly faster on both Intel and AMD processors.*
Keywords: *Machine learning, Intel Extension for SciKit-Learn, Earth observation, Bare soil detection*

1. Introduction

Scikit-learn, a Python library, is broadly adopted in machine learning (ML) applications [1]. It offers a wide range of ML algorithms, making it accessible to non-specialists through a uniform Python interface. Its focus on usability, performance, documentation, and API consistency makes it a go-to tool for both academic and commercial environments [2]. It is used across various sectors (by e.g., Spotify

DOI:10.17388/WUT.2024.0002.MiNI

or Hugging Face) due to its versatility and comprehensive range of (un)supervised ML algorithms, and Scikit-learn is one of the most widely used ML libraries in Python [2], with over 12 million weekly downloads on GitHub back in 2019.

Since the era of Big Data is here, we need to face the challenges concerned with the volume, velocity, variety, and veracity of data generated daily in practically all domains of science and industry. Therefore, building rapid training and inference pipelines is of paramount importance, especially if ML models are to be deployed on board edge devices with hardware constraints. Recently, an Intel® Extension for the Scikit-learn package has been introduced, promising to significantly speed up training and inference through leveraging the abilities of processors that are equipped with at least one of the SSE2, SSE4.2, AVX2, AVX512 instruction sets [3]. We investigate its capabilities in a real-world Earth observation downstream task of bare soil detection (Section 2), which is pivotal in automated soil analysis from satellite imagery [4]. The results of our computational experiments indicate that exploiting the Intel® Extension for the Scikit-learn may bring significant improvements for not only Intel but also AMD processors (Section 3).

2. Materials and Methods

We discuss the bare soil detection benchmark in Section 2.1. The investigated machine learning models and the experimental settings are detailed in Section 2.2.

2.1. Bare Soil Detection Dataset

In this study, we focus on a dataset of airborne hyperspectral images (HSIs) accompanied by manual ground-truth bare soil delineations. Hyperspectral imagery captures up to hundreds of narrow and contiguous spectral bands within a given spectral range, offering very detailed information about the scanned materials. HSIs are, however, highly dimensional and thus difficult to transfer, store, and analyze in the case of spatially large areas. The dataset used in this study was captured during the campaign in the Lower Silesia region, Poland, in March 2021 [5]. The data acquisition was facilitated by two sensors, with a spectral range of 930–2500 nm (5.45 nm spectral resolution, 288 spectral bands), and with a spectral range of 400–1000 nm (3.26 nm resolution, 186 bands)—the resulting HSI contained 430 spectral bands and the Ground Sample Distance (GSD) of 2 m. This HSI (of size 4344×7020×430, with the latter dimension corresponding to the number of spectral bands), was split into non-overlapping segments of

120×120×430—see an example patch rendered in Figure 1.

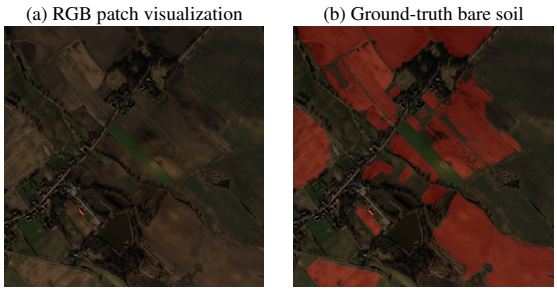


Figure 1. Example (a) RGB visualization of a patch from the investigated dataset, together with (b) the ground-truth bare soil pixels (annotated in light red).

The resulting 1446 patches were stratified into $k = 5$ folds using k-fold cross-validation (CV), considering the pixel count corresponding to bare soil in each image patch. We target the bare soil detection problem—since exploiting RGB images may not be enough here, especially for the areas with early stages of vegetation or urban objects, we aim at benefiting from much more detailed HSIs. In this study, we randomly sampled 10^5 hyperspectral pixels that were accompanied by ground-truth class labels (bare soil vs. background).

2.2. Machine Learning Models and Experimental Design

As feature vectors, we exploit 150 spectral bands that were used to initially emulate the Intuition-1 characteristics, as suggested in [5]. In this study, we investigated the following configuration aspects (each experiment was repeated 10×):

- **7 classifiers**: random forest (RF), k-nearest neighbors (kNN), support vector classifiers, without and with controlling the number of support vectors (SVC and NuSVC), and logistic regression (LR), together with the classifiers that are not officially supported by the Intel Extension: a linear model with L_2 regularization (L2), and linear models with stochastic gradient descent (SGD) learning.
- **3 datasets variants**, with 10^5, 10^4, and 10^3 training samples (N). The latter two sets are randomly sampled from the whole set of 10^5 examples, to investigate the impact of the training set size of both functional (F_1) and non-functional (training and inference times) capabilities of the models.

- **3 hardware configurations**: Intel i7-12700H 2300MHz with 14 cores and 20 logical processors, Intel® i7-8565U 1992MHz with 4 Cores and 8 logical processors, and AMD® Ryzen 7 3700x with 8 cores 16 logical processors.

To entire codebase used in this study is publicly available [1], and the Intel Extension [3] can be installed via Python package managers.

Table 1. The experimental results, with mean and its standard deviation. If the time was decreased at least 2× with the use of Intel Extension, it is **boldfaced**.

	training time $[10^{-3}s]$		**inference time** $[10^{-3}s]$		F_1	
N	Sklearn	Intel	Sklearn	Intel	Sklearn	Intel
k-Nearest Neighbors						
10^3	—	—	24.9±23.3	**1.5**±0.4	0.90±0	0.90±0
10^4	—	—	131±2.6	**35.5**±2.3	0.88±0	0.88±0
10^5	—	—	5116±46	**1831**±19	0.90±0	0.90±0
LogisticRegression						
10^3	502±144	**5.7**±0.4	1.4±0.1	**0.3**±0.0	0.85±0	0.85±0
10^4	982±683	**32.7**±2.0	3.5±0.7	**0.7**±0.0	0.86±0	0.86±0
10^5	1726±274	**499**±10	10.7±0.7	**2.5**±0.1	0.87±0	0.74±0
SVC						
10^3	20.5±0.5	11.7±0.6	2.6±0.1	**0.7**±0.1	0.85±0	0.85±0
10^4	2149.3±29.5	**358.9**±2.8	251.6±0.9	**11.3**±0.2	0.86±0	0.86±0
10^5	450578±10965	**13678**±207	44385±2402	**913**±13	0.87±0	0.87±0
NuSVC						
10^3	22.2±0.8	**9.1**±0.4	3.0±0.3	**0.7**±0.1	0.86±0	0.87±0
10^4	3022.9±47.9	**316.4**±2.1	254.6±0.4	**11.5**±0.2	0.86±0	0.86±0
10^5	899998±674	**11927**±39	41130±242	**904**±3.7	0.86±0	0.87±0
Random Forest						
10^3	434.6±3.5	**14.3**±0.7	56.1±8.1	**1.8**±0.3	0.92±0	0.92±0
10^4	2139.5±8.7	**217.5**±2.0	72.9±8.8	**8.7**±0.7	0.88±0	0.88±0
10^5	25807±112	**4245**±89	170.3±5.5	**99.9**±1.3	0.91±0	0.91±0
Linear model with L_2 regularisation						
10^3	10.4±1.4	**3.9**±1.5	1.0±0.1	**0.3**±0.1	0.72±0	0.72±0
10^5	130.4±11.3	122.5±9.0	3.6±0.3	2.6±0.3	0.78±0	0.78±0
Stochastic Gradient Descent						
10^3	10.4±2.9	9.7±2.0	1.4±0.1	1.3±0.2	0.79±0.07	0.79±0.07
10^5	320.6±19.5	329.8±19.7	3.5±0.2	2.6±0.3	0.87±0.01	0.87±0.01

3. Experimental Results

Table 1 gathers the experimental results, with and without using the Intel Extension (all other experimental settings remain unchanged). To verify the classification abilities of the models, we also report the F_1 score. The main outcome of the evaluation is that the duration of training and inference processes decreased

significantly after employing the Intel Extension, without any loss in classification performance. The reference models that have not yet been optimized (L2 and SGD), as expected, did not display considerable change in measured training and inference times. Additionally, we calculated $r_{\Theta,i} = \Theta_i^{SciKit-Learn}/\Theta_i^{Intel}$, where Θ_i is a training/inference time of an *i-th* model. Thus, $r_\Theta > 1$ indicates the accelerated training/inference when compared to its counterpart without the Intel Extension (Figure 2). There are models with significant improvements for inference (LR, RF), whereas the others had their training notably accelerated—see e.g., SVCs, suffering from significant time and memory complexity of this process [6]. We can appreciate the best improvements for LR (even 100× faster inference, $N = 10^3$) and NuSVR (75×, $N = 10^5$). Very popular RFs were accelerated as well. Of note, the improvements were obtained not only on the (newest) Intel platforms but also on their older environments, as well as over the AMD processors (Figure 3).

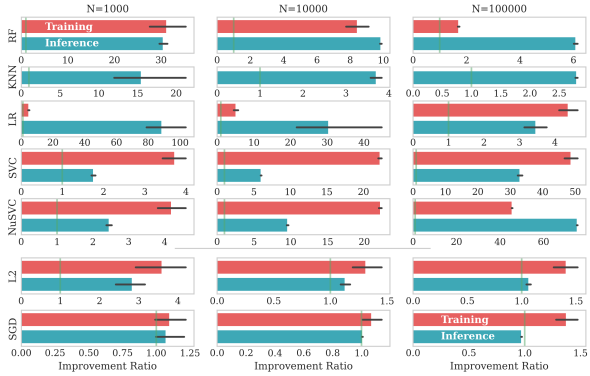


Figure 2. Time improvements for all models and training set sizes (N).

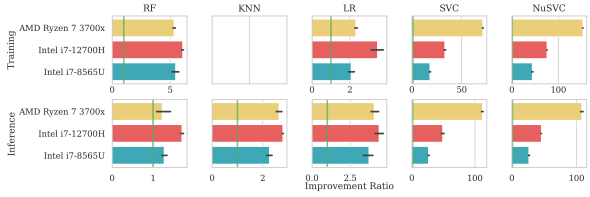


Figure 3. Time improvements for all execution environments.

4. Conclusions

Accelerating machine learning models is of pivotal importance in the world of Big Data. In this work, we investigated the capabilities of the recent Intel Extension for a popular Scikit-learn machine learning package over a real-world benchmark dataset, targeting the bare soil detection problem from highly-dimensional hyperspectral imagery. The experiments revealed that the extension offers significant acceleration of both training and inference, not only over the Intel platforms but also over the AMD processors. Researching the potential profits of using this technique (which is independent of the underlying downstream classification or regression task) on other hardware platforms may bring exciting opportunities in on-board artificial intelligence [7], where such models are deployed on edge devices, e.g., Earth observation satellites with compute and memory constraints.

References

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

[2] Chary, D. et al. Review on advanced machine learning model: Scikit-Learn. *IJSRED*, 3(4):526–529, 2020. URL `ssrn.com/abstract=3694350`.

[3] Intel® extension for Scikit-learn, 2023. URL `intel.github.io/scikit-learn-intelex`.

[4] Wijata, A. M. et al. Taking artificial intelligence into space through objective selection of hyperspectral earth observation applications: To bring the "brain" close to the "eyes" of satellite missions. *IEEE Geosci. Remote Sens. Mag.*, 11(2):10–39, 2023.

[5] Nalepa, J. et al. The hyperview challenge: Estimating soil parameters from hyperspectral images. In *Proc. IEEE ICIP*, pages 4268–4272. 2022.

[6] Nalepa, J. and Kawulok, M. Selecting training sets for support vector machines: a review. *Artificial Intell. Review*, 52(2):857–900, 2019.

[7] Ruszczak, B. et al. Machine learning detects anomalies in ops-sat telemetry. In *Computational Science – ICCS 2023*, pages 295–306. Springer Nature, 2023.

Unsupervised Anomaly Detection in Time Series with Evolving Spiking Neural Networks and Empirical Mode Decomposition

Mariusz Paluch, Piotr Stanisław Maciąg[0000−0001−5486−7927]

Warsaw University of Technology
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-661 Warsaw, Poland
piotr.maciag@pw.edu.pl

Abstract. *We propose an ensemble of unsupervised anomaly detectors using evolving spiking neural networks (eSNNs) and time series decomposition. Initially, a univariate time series is decomposed into components via empirical mode decomposition. An ensemble of eSNNs then independently detects anomalies in the original series or its components. Anomaly detection at any time point is determined through majority or weighted voting strategies. Tested with the NAB repository, our experiments indicate this ensemble enhances anomaly detection quality in certain time series categories.*
Keywords: *spiking neural networks, anomaly detection, time series decomposition*

1. Introduction

Anomaly detection in time series data is a very important research topic that has been studied over many years and in numerous publications (see, for example, [1, 2, 3]). In this work, we present the results suggesting that by applying empirical mode decomposition of a time series, we can enhance anomaly detection results in this time series data. Specifically, in the article, we have tested the Complete Ensemble Empirical Mode Decomposition With Adaptive Noise (CEEMDAN) algorithm. The selected algorithm decomposes time series X into a set of Intrinsic Mode Functions (IMFs) or modes. The common characteristic of each IMF includes two conditions [4]: (i) the number of extrema and the number of zero

DOI:10.17388/WUT.2024.0002.MiNI

crossings must be equal or differ at most by one; and (ii) the mean value of the upper and lower envelope is zero everywhere. The state-of-the-art CEEMDAN algorithm improves other empirical mode decomposition algorithms, such as EMD or EEMD.

Previous research suggests that time series decomposition can improve the quality of anomaly detection (see, for example, [5, 6]). However, to the best of our knowledge, for the first time, we applied an Empirical Mode Decomposition algorithm in order to improve the quality of anomaly detection in time series data. Other recent results also suggest that Online Evolving Spiking Neural Networks (OeSNNs) are highly effective in time series processing, and especially unsupervised anomaly detection in time series/data streams ([7, 2, 8]). Thus, in this research, we decided to combine both approaches in order to offer a new, ensemble approach to anomaly detection.

In Section 2 we describe the designed architecture of the ensembles of detectors along with the decomposition model. Section 3 provides the results of experiments. Finally, in Section 4 we give conclusions of the work and provide further research directions.

2. The Proposed Ensemble Architecture

The proposed ensemble method is presented in Figure 1. The input time series is decomposed using the CEEMDAN method into a series of modes. In the ensemble-based approach, a set of detectors is employed to identify anomalies within both the original time series and its decomposed components. Each series, whether it be the original or a component, is analyzed by its dedicated network within the ensemble. The final determination of whether an anomaly is present at any given time point is made through a consensus mechanism, which can either be majority voting, where each network has an equal influence on the outcome, or weighted voting, where the contribution of each network is adjusted based on its performance or reliability. This method leverages the collective insight from multiple perspectives to enhance the accuracy and robustness of anomaly detection.

Let us assume that time series is denoted as $X = [x_1, x_2, \ldots, x_t, \ldots, x_T]$. Let us assume that for time point t, we have a set of detections D_t of N detectors in the ensemble. $D_t = [d_{0,t}, d_{1,t}, \ldots, d_{N,t}]$, where each $d_{i,t} \in \{0, 1\}$ and 0 indicates the absence of an anomaly, while 1 indicates the presence of an anomaly. Specifically, the two voting strategies employed in the ensemble are as follows:

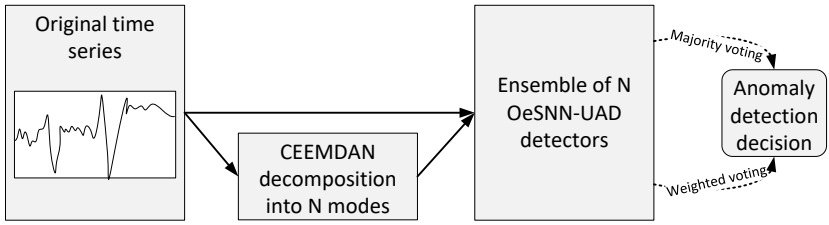


Figure 1. The architecture of the proposed ensemble model.

- The first strategy is simple majority voting. At time point t, an anomaly is reported by the ensemble when $|d \in D_t : d = 1| > \frac{|D_t|}{2}$.
- The second strategy employs weighted voting. At time point t, an anomaly is reported by the ensemble when $\sum_{i=0}^{N} w_i \cdot d_{i,t} > \frac{\sum_{i=0}^{N} w_i}{2}$, in which w_i is a weight assigned to each network in the ensemble and calculated as the square of the reversed index of the CEEMDAN component processed by a network, given that the original time series is treated as the first component.

In the proposed architecture, we applied the OeSNN-UAD detector that was proposed in [2]. In the experiments, we used the OeSNN-UAD implementation prepared in Python and publicly available as a PIP package `https://pypi.org/project/OeSNN-AD/`.

3. Experiments

The proposed ensemble was tested using time series data from one publicly available repository: the Numenta Anomaly Benchmark (NAB) [9]. The NAB dataset offers a comprehensive environment for evaluating anomaly detection algorithms with over 50 real-world and artificial time series datasets, employing a scoring system that rewards timely and accurate detection. NAB emphasizes a variety of streaming data contexts with a unique scoring mechanism. However, it is worth to note that at least some time series of the repository posses some drawbacks, such as incorrect anomaly labeling or trivial anomalies (see, for example, [10]). The results presented beneath were obtained for 5 CEEMDAN components (excluding the original time series).

In order to evaluate anomaly detection with the proposed ensemble, we applied F1-measure. The formula for the measure is provided in Eq. (1).

$$\text{Prec.} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|}, \text{Rec.} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|}, \text{F1} = 2 \cdot \frac{\text{Prec.} \cdot \text{Rec.}}{\text{Prec.} + \text{Rec.}}. \tag{1}$$

Beneath, we present the results of the experiments. The results for the detectors other than OeSNN-UAD are reported in [11]. Both for the singleton OeSNN-UAD detector as well as for the proposed ensemble, we conducted a grid-search procedure in order to find the parameters giving the best results for the range of parameters specified in [2].

Table 1. Average F1-measure values obtained for NAB using the selected detectors (marked with *) presented in [11] and using the OeSNN-UAD ensemble with weighted voting. (**The ensemble OeSNN-UAD applies weighted voting strategy.)

Dataset category	*Skyline**	*Twitter ADVec**	*Windowed Gaussian**	*DeepAnT**	*Single OeSNN-UAD*	*Ens. OeSNN-UAD***
Artificial with Anomaly	0.043	0.017	0.013	0.156	**0.517**	0.344
Real Ad Exchange	0.005	0.018	0.026	0.132	0.267	**0.352**
Real AWS Cloud	0.053	0.013	0.06	0.146	**0.458**	0.391
Real Known Cause	0.008	0.017	0.006	0.2	**0.347**	0.310
Real Traffic	0.091	0.020	0.045	0.223	0.318	**0.395**
Real Tweets	0.035	0.018	0.026	0.075	**0.284**	0.281

As noted in Table 1, the ensemble of OeSNN-UAD detectors can improve anomaly detection in some categories of time series when the weighted voting strategy is applied. Generally, when the majority voting strategy was applied in the ensemble, detection for only specific time series proved to be superior to that of the singleton OeSNN-UAD. In Table 2, we list these time series.

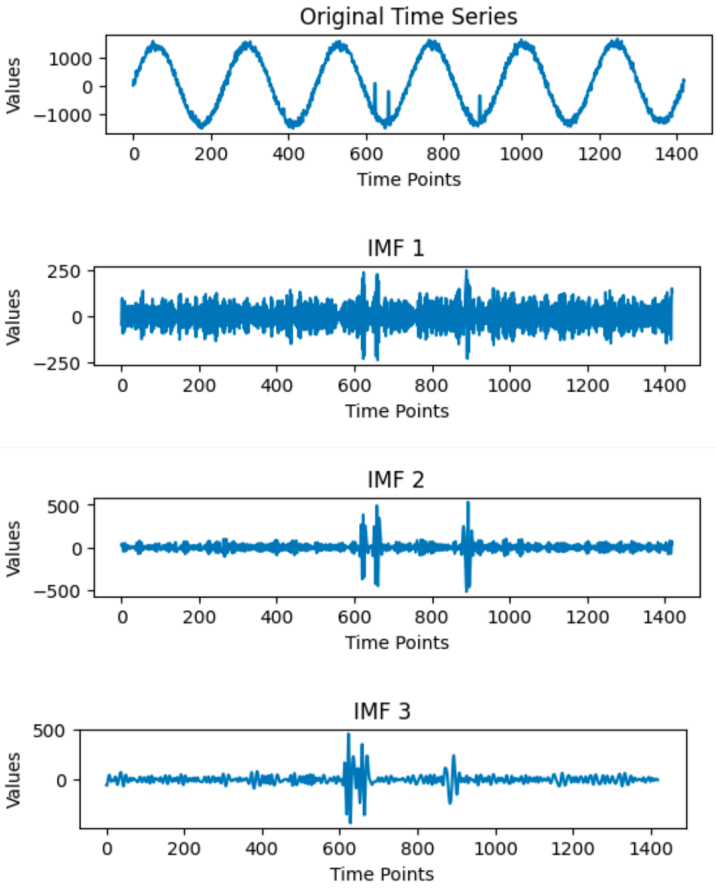


Figure 2. Example of original time series data with present anomalies and its first three IMFs obtained using the CEMMDAN decomposition. It can be clearly noted that three anomalies present (around time points 600, 650 and 900) in the original time series are also distinctively represented in the obtained modes.

The conducted experiments suggest that the weighted voting strategy is more effective in anomaly detection than majority voting. This effectiveness can be

Table 2. The time series for which the ensemble of OeSNN-UADs involving majority voting provided better results than a singleton OeSNN-UAD detector. (*Ensemble OeSNN-UAD applies majority voting strategy in the result beneath.)

Dataset category	OeSNN-UAD	Ens. OeSNN-UAD*
occupancy_t4013	**0.517**	0.344
speed_t4013	0.267	**0.352**
exchange-4_cpc_results	**0.458**	0.391
Twitter_volume_AAPL	**0.347**	0.310
ambient_temperature_system_failure	0.318	**0.395**

attributed to the fact that each subsequent mode disturbs the original time series more significantly. Such distortion can lead to the misdetection of anomalies when each obtained mode is treated equally.

4. Conclusions and Future Work

In this paper, we presented a new approach to unsupervised anomaly detection in time series based on the CEEMDAN decomposition and an ensemble of online evolving spiking neural networks. To test the ensemble, we utilized a dataset from the NAB repository, which includes more than 50 time series. Experiments suggest that our proposed approach can surpass the anomaly detection quality of other state-of-the-art detectors in some categories of datasets when combined with the weighted voting strategy. To fully verify the usefulness of the ensemble, experiments with other anomaly detection repositories are necessary. Furthermore, future studies should also compare the effectiveness of the applied time series decomposition method in anomaly detection against other methods, e.g., the FFT method. Another potential development for the ensemble could involve a more sophisticated decision-making process during decomposition. For instance, the initial decision on the presence of an anomaly could be made solely based on the original time series, and then, the obtained modes could be used to verify whether the anomaly should be reported.

References

[1] Ahmad, S., Lavin, A., Purdy, S., and Agha, Z. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134 – 147, 2017. Online Real-Time Learning Strategies for Data Streams.

[2] Maciąg, P. S., Kryszkiewicz, M., Bembenik, R., L. Lobo, J., and Del Ser, J. Unsupervised anomaly detection in stream data with online evolving spiking neural networks. *Neural Networks*, 139:118–139, 2021. ISSN 0893-6080.

[3] Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3), 2021. ISSN 0360-0300. doi:10.1145/3444690. URL `https://doi.org/10.1145/3444690`.

[4] Torres, M. E., Colominas, M. A., Schlotthauer, G., and Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4144–4147. 2011. doi:10.1109/ICASSP.2011.5947265.

[5] Choudhary, S., Hiranandani, G., and Saini, S. K. Sparse decomposition for time series forecasting and anomaly detection. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 522–530. SIAM, 2018.

[6] Zhang, C., Zhou, T., Wen, Q., and Sun, L. Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 2497–2507. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450392365. doi: 10.1145/3511808.3557470. URL `https://doi.org/10.1145/3511808.3557470`.

[7] Maciąg, P. S., Kryszkiewicz, M., and Bembenik, R. Online evolving spiking neural networks for incremental air pollution prediction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. 2020.

[8] Bäßler, D., Kortus, T., and Gühring, G. Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks. *Machine Learning*, 111(4):1377–1408, 2022.

[9] Ahmad, S., Lavina, A., Purdya, S., and Agha, Z. Numenta anomaly benchmark, 2017. URL `https://github.com/numenta/NAB`.

[10] Wu, R. and Keogh, E. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[11] Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2019. ISSN 2169-3536.

Selected properties of Grid Graph Laplacians

Mieczysław A. Kłopotek[1][0000−0003−4685−7045],
Sławomir T. Wierzchoń[2][0000−0001−8860−392X]

[1]*Institute of Computer Science, Polish Academy of Sciences*
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland, klopotek@ipipan.waw.pl
[2]*Institute of Computer Science, Polish Academy of Sciences*
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland, stw@ipipan.waw.pl

Abstract. *The paper investigates some properties of the regular multidimensional grid graphs from the point of view of the analytical foundations for studying the clustering behavior of the GSA (Graph Spectral Analysis) clustering algorithms. From the theoretical point of view, the regular grid graph has no structure, and therefore the properties of spectra of the corresponding Laplacians may be indicative of a missing structure in the data. The range of eigenvalues is determined and it is shown that the eigenvalues are not distributed uniformly. This contradicts assumptions made in many GSA applications. The complexity of eigenvector elements distributions is shown.*
Keywords: *Graph Spectral Analysis, multidimensional regular grid graphs, combinatorial Laplacians, analytical eigen-decomposition*

1. Introduction

Regular graph structures and their properties attract much attention due to, among others, analytical derivation of graph properties [1]. Stankiewicz [2] investigates the impact of the minimum number of handles to be added to the plane to embed this graph without crossings on the spectrum of its Laplacian. Cornelissen et al. [3] investigate gonality of curves using grid Laplacians. Cheung et al. [4] is interested in grid structures when creating applications in image processing. Merris [5] exploits properties of grid graph Laplacians in chemical applications. Cetkovic et al. [6] study their applications in mechanics (membrane vibration).

We have investigated such graphs mainly from the point of view of properties of Graph Spectral Clustering algorithms [7, 8, 9]. The study of regular grids is useful for the validation of some assumptions made for algorithms clustering via GSA large datasets, like Laplacian eigenvalue uniformity.

DOI:10.17388/WUT.2024.0002.MiNI

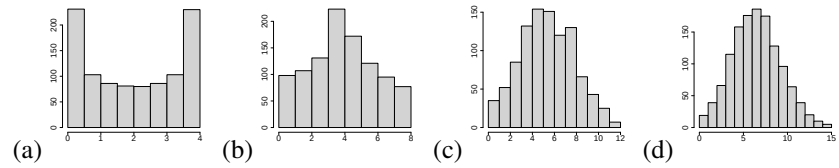


Figure 1. The histograms of eigenvalues of combinatorial Laplacians of grid graphs of approximately 1,000 nodes. (a) 1-dimensional grid graph, (b) 2-dimensional grid graph, (c) 3-dimensional grid graph, (d) 4-dimensional grid graph.

2. The Problem

Let us briefly recall the basic mechanism underlying clustering via GSA. One starts with the similarity matrix of objects S. Then one computes e.g. the combinatorial Laplacian (that we are interested in here) as $L = D - S$, where D is a diagonal matrix with components being row sums of S. The eigenvectors corresponding to the lowest positive eigenvalues of L are used as an embedding of objects in a low-dimensional Euclidean space where clustering is performed using e.g. k-means algorithm. See e.g. [10, 11] for details.

We study here regular grid graphs of 1 to d dimensions where in each dimension a different number of layers can exist. To be more formal define the path graph P_n as a tree with two nodes of vertex degree 1, and the other $n-2$ nodes of vertex degree 2. Then a graph $G(n_1, \ldots, n_d)$ is called a d-dimensional grid if it is the Cartesian product of d paths $P_{n_1}, \ldots, , , P_{n_d}$, [12]. Such a graph can be embedded into an d dimensional Euclidean space with nodes having only consecutive integer coordinates. The nodes of a grid graph are linked with coordinates differing only by 1 at only one axis. Those graphs are of special interest because they can be used for practical implementation of parallel algorithms. In the similarity matrix, the entry is equal to 1, if there is a link between nodes.

We have shown [7] that for such a regular grid structure, the combinatorial Laplacian eigenvalues λ and eigenvectors $\mathbf{v}$ can be derived as follows. Define

$$\lambda_{[z_1,\ldots,z_d]} = \sum_{j=1}^{d} \left(2 \sin\left(\frac{\pi z_j}{2n_j} \right) \right)^2 \tag{1}$$

where, for each $j = 1, \ldots, d$, z_j is an integer such that $0 \leq z_j \leq n_j - 1$. Denoting

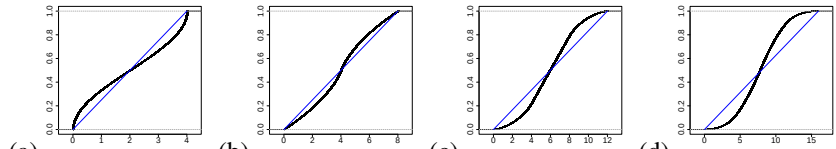


Figure 2. The in the limit cumulative distribution function of eigenvalues of combinatorial Laplacians of grid graphs. (a) 1-dimensional grid, (b) 2-dimensional grid, (c) 3-dimensional grid, (d) 4-dimensional grid graph. Blue line – the uniform distribution function

$\lambda_{(j,z_j)} = \left(2\sin\left(\frac{\pi z_j}{2n_j}\right)\right)^2$ the above formula can be rewritten as $\lambda_{[z_1,\ldots,z_d]} = \sum_{j=1}^{d} \lambda_{(j,z_j)}$. Define furthermore

$$\nu_{[z_1,\ldots,z_d],[x_1,\ldots,x_d]} = \prod_{j=1}^{d} \cos\left(\frac{\pi z_j}{n_j}\left(x_j - 0.5\right)\right) \tag{2}$$

where, for each $j = 1,\ldots,d$, x_j is an integer such that $1 \le x_j \le n_j$.

And finally define the n dimensional vector $\mathbf{v}_{[z_1,\ldots,z_d]}$ such that

$$\mathbf{v}_{[z_1,\ldots,z_d],i} = \nu_{[z_1,\ldots,z_d],[x_1,\ldots,x_d]} \tag{3}$$

Theorem 1. *If* $z_j \in [-n_j + 1, -1]$, *then* $\mathbf{v}_{[z_1,\ldots,z_j,\ldots,z_d]} = \mathbf{v}_{[z_1,\ldots,z'_j,\ldots,z_d]}$ *where* $z'_j \in [0, n_j - 1]$, *and* $z'_j = -z_j$. *If* $z_j = n_j$, *then* $\mathbf{v}_{[z_1,\ldots,z_j,\ldots,z_d]} = \mathbf{0}$. *If* $z_j \in [n_j + 1, 2n_j - 1]$, *then* $\mathbf{v}_{[z_1,\ldots,z_j,\ldots,z_d]} = -\mathbf{v}_{[z_1,\ldots,z'_j,\ldots,z_d]}$ *where* $z'_j \in [0, n_j - 1]$, *and* $z'_j = 2n_j - z_j$.

In the subsequent section we look at some properties of these sets of eigenvectors and eigenvalues when the number of nodes in the regular graph grows.

3. Some Properties of Combinatorial Laplacian for a multidimensional grid graph

The combinatorial Laplacian of a grid graph, according to the formula (1), ranges from 0 to 2^d, where d is the dimensionality of the grid graph. When the lowest number of layers in any dimension grows to infinity, the upper bound on the highest λ is approached. Many approximate methods for GSA clustering, like compressive spectral clustering CSC [13], assume that the eigenvalues of

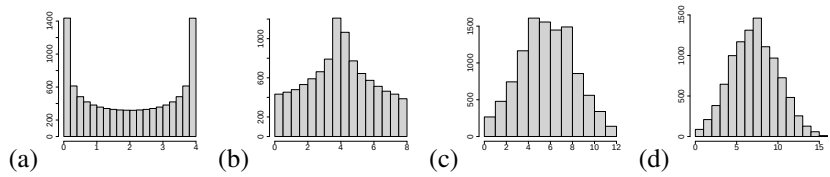


Figure 3. The histograms of eigenvalues of combinatorial Laplacians of grid graphs of approximately 10,000 nodes. (a) 1-dimensional grid graph, (b) 2-dimensional grid graph, (c) 3-dimensional grid graph, (d) 4-dimensional grid graph.

the Laplacian are uniformly distributed. A glance at the formula (1) denies this assumption for combinatorial Laplacians. This is also visible from visual inspection.[1] You see the histograms of eigenvalue for grid graphs of approximately 1,000 nodes with dimensionality ranging between 1 and 4 in Figure 1. Figure 3 depicts analogous histograms for 10,000 node graphs. The shapes of histograms for 1,000 nodes and 10,000 nodes are similar for the same grid dimensionality and they are in no way uniform (as we see for 1 to 4-dimensional grid graphs).

Cumulative distribution functions of eigenvalues of combinatorial Laplacians of 1, 2, 3, and 4-dimensional grid graphs with the number of nodes "in the limit" are depicted in Figure 2. A blue line was added in each diagram to indicate how a uniform distribution would have looked like. The multidimensional grid graph exhibits no similarity to uniform eigenvalue distribution though it is structureless.

Figure 4 visualizes the shapes of sample eigenvectors of the aforementioned grids. A clear cosine shape can be recognized for a 1-dimensional graph. One can recognize the cosine product for two-dimensional grids. However, the patterns are not so easily classified by eye inspection in higher dimensions. This behavior questions assumptions of some GSA on piece-wise linearity of eigenvector entries.

4. Conclusions

Shapes of eigenvalue spectra seem to be kept with growing number of nodes and they are in no way uniform. Such a study should be a clear warning sign for

[1] We revealed these properties in a broader context in an unpublished manuscript [8], but we feel that they deserve greater attention as the uniformity assumptions are made over and over again.

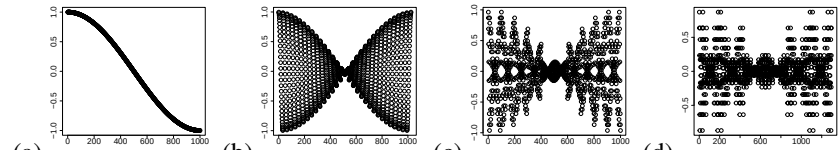


Figure 4. The plots of sample eigenvectors of combinatorial Laplacians of grid graphs of approximately 1,000 nodes. (a) 1-dimensional grid graph, $\mathbf{z} = [1]$, (b) 2-dimensional grid graph, $\mathbf{z} = [1, 1]$, (c) 3-dimensional grid graph, $\mathbf{z} = [1, 1, 1]$, (d) 4-dimensional grid graph, $\mathbf{z} = [1, 1, 1, 1]$.

those who develop approximate procedures for clustering large scale graphs, like CSC [13], as the common assumption of uniformity is not valid.

Entries in eigenvectors form a clear cosine line for 1-dimensional graph, but shapes in higher dimensions do not seem to be obvious. This should be a concern when utilizing GSA methods for large graphs like Nyström like approaches, [14].

References

[1] Notarstefano, G. and Parlangeli, G. Controllability and observability of grid graphs via reduction and symmetries. CoRR abs/1203.0129, 2012.

[2] Stankewicz, J. On the gonality, treewidth, and orientable genus of a graph. CoRR abs/1704.06255, 2017.

[3] Cornelissen, G., Kato, F., and Kool, J. A combinatorial Li-Yau inequality and rational points on curves. *Math. Ann.*, 361(1):211–258, 2015.

[4] Cheung, G., Magli, E., Tanaka, Y., and Ng, M. K. Graph spectral image processing. *Proceedings of the IEEE*, 106(5):907–930, 2018.

[5] Merris, R. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197:143–176, 1994. ISSN 0024-3795. doi:10.1016/0024-3795(94)90486-3.

[6] Cvetković, D. M., Doob, M., and Sachs, H. *Spectra of graphs: theory and application*. Academic Press, 1980.

[7] Kłopotek, M. A., matrices of graphs: a surveyławomir T. Wierzchoń, S., and Kłopotek, R. A. Analytical forms of normalized and combimnatorial laplacians of grid graphs. In *Proc. PP-RAI'2019*, pages 281–284. 2019.

[8] Kłopotek, M. A. Spectral Analysis of Laplacian of a Multidimensional Grid Graph - Combinatorial versus Normalized and Random Walk Laplacians. *ArXiv e-prints 1707.05210*, 2019.

[9] Kłopotek, M. A., Wierzchoń, S. T., and Kłopotek, R. A. Weighted laplacians of grids and their application for inspection of spectral graph clustering methods. *TASK Quarterly*, 25(3):329–353, 2021. doi:10.34808/tq2021/25.3/d.

[10] von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[11] Wierzchoń, S. T. and Kłopotek, M. A. *Modern Clustering Algorithms*, volume 34 of *Studies in Big Data*. Springer Verlag, 2018. ISBN 978-3-319-69307-1.

[12] Spielman, D. Spectral and algebraic graph theory. Book draft. http://cs-www.cs.yale.edu/homes/spielman/sagt, 2019.

[13] Tremblay, N., Puy, G., Gribonval, R., and Vandergheynst, P. Compressive spectral clustering. In *Proc. of the 33rd Intl. Conf. on Machine Learning*, volume 48 of *ICML'16*, pages 1002–1011. JMLR.org, 2016.

[14] Li, M., Bi, W., Kwok, J. T., and Lu, B.-L. Large-scale Nyström kernel matrix approximation using randomized svd. *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):152–164, 2015. doi:10.1109/TNNLS.2014.2359798.

Meta-learning for combining forecasts: deterministic and probabilistic approaches

Grzegorz Dudek[0000−0002−2285−0327]

Częstochowa University of Technology
Faculty of Electrical Engineering
Armii Krajowej 17, 42-200 Częstochowa, Poland
grzegorz.dudek@pcz.pl

Abstract. *In this research, we delve into the realm of meta-learning to combine forecasts generated by individual base models for both deterministic and probabilistic forecasting. For deterministic forecasting, we explore a range of meta-learners, including linear regression, k-nearest neighbors, multilayer perceptron, random forest, and long short-term memory. For probabilistic forecasting, we leverage quantile linear regression, quantile regression forest, and post-processing technique involving residual simulation. We introduce both global and local meta-learning variations specifically designed for time series data marked by complex seasonality patterns.*
Keywords: *ensemble forecasting, meta-learning, multiple seasonality, probabilistic forecasting, stacking*

1. Introduction

Ensembling has proven to be remarkably effective in bolstering the predictive capabilities of forecasting models. Using different combining mechanisms, ensembling mitigates concerns surrounding base model structure and parameter specification [1]. Consequently, it diminishes the risk associated with relying solely on the limitations or biases inherent in a single model, providing a more comprehensive depiction of the underlying data generation process.

While averaging and linear regression are the most prevalent methods for combining forecasts, they may fall short when the relationship between forecasts generated by the base models and the target value is nonlinear and exhibits complex

DOI:10.17388/WUT.2024.0002.MiNI

patterns. In such scenarios, machine learning (ML) models offer an alternative approach by leveraging stacking procedures [2, 3]. Stacking enhances forecast accuracy by learning the optimal combination of constituent forecasts in a data-driven manner.

The aim of this study is to evaluate a range of ML models employing diverse strategies to tackle the challenge of combining base forecasts, yielding either point or probabilistic forecasts. We investigate five ML models for deterministic forecasting and three methods for probabilistic forecasting. Within our stacking framework, we explore two training strategies: global and local. The global approach utilizes all available historical data for training the meta-models, while the local approach focuses on training patterns most similar to the query pattern.

2. Global and Local Meta-Learning

The meta-model aggregates point forecasts for time t generated by n forecasting base models to create either point or probabilistic forecasts. In this study, we generate probabilistic forecasts in the form of a set of quantiles. The base forecasts are represented by vector $\hat{\mathbf{y}}_t = [\hat{y}_{1,t}, ..., \hat{y}_{n,t}]$. The combined point forecast is denoted as $\tilde{y}_t = f(\hat{\mathbf{y}}_t; \boldsymbol{\theta}_t)$, where $\boldsymbol{\theta}_t$ represents the meta-model parameters. The combined quantile forecast is given as $\tilde{\mathbf{q}}_t = f(\hat{\mathbf{y}}_t; \boldsymbol{\theta}_t)$, where $\tilde{\mathbf{q}}_t = [\tilde{q}_t(\alpha)]_{\alpha \in \Pi}$, and Π represents the assumed set of probabilities $\alpha \in [0, 1]$.

The training set for both deterministic and probabilistic meta-models is the same: $\Phi = \{(\hat{\mathbf{y}}_\tau, y_\tau)\}_{\tau \in \Xi}$, where y_τ denotes the target value, Ξ is a set of selected time indices from the interval $T = 1, ..., t - h$, and h is the forecast horizon. In the global mode, the meta-model is trained using all available historical data from period $\Xi = \{1, ..., t - h\}$. This allows the model to leverage all past information to generate a forecast for the current time point t.

In the local mode, the goal is to train the meta-model locally around the query pattern $\hat{\mathbf{y}}_t$. To achieve this, k most similar input vectors to $\hat{\mathbf{y}}_t$ are selected and included in the local training set. This approach is applied to non-recurrent meta-models, using the Euclidean metric to determine the nearest neighbors. However, for recurrent models such as long short-term memory (LSTM), which capture dynamics from sequential data, this approach disrupts the temporal structure of data. To address this, local learning for LSTM is defined in three ways as follows:

v1 The training sequence is restricted to the last c points, i.e. $\Xi = \{t-h-c, ..., t-h\}$, allowing LSTM to model the relationship for the query pattern based on

the most recent sequence of length c.

v2 For seasonal time series, it may be beneficial to select training points that are lagged to the forecasted point by the length of the seasonal period s_1: $\Xi = \{t - cs_1, t - (c-1)s_1, ..., t - s_1\}$. This training set preserves the time structure but disregards the seasonal pattern, only including points in the same phase of the seasonal cycle as the forecasted point.

v3 In the case of series with double seasonality (s_1 and s_2, where s_2 is a multiple of s_1), a modified training set contains points from the same phase of both seasonal patterns as the forecasted point: $\Xi = \{t - cs_2, t - (c-1)s_2, ..., t - s_2\}$.

3. Meta-Models

3.1. Deterministic Forecasting [4]

LR Linear Regression: $f(\hat{\mathbf{y}}) = \sum_{i=1}^{n} a_i \hat{y}_i + a_0$, where $a_0, ..., a_n$ are coefficients.

kNN k-Nearest Neighbours: $f(\hat{\mathbf{y}}) = \frac{\sum_{\tau \in \Xi} w(\hat{\mathbf{y}}, \hat{\mathbf{y}}_\tau) y_\tau}{\sum_{\tau \in \Xi} w(\hat{\mathbf{y}}, \hat{\mathbf{y}}_\tau)}$, $w(\hat{\mathbf{y}}, \hat{\mathbf{y}}_\tau) = \exp\left(-\frac{\|\hat{\mathbf{y}} - \hat{\mathbf{y}}_\tau\|^2}{\sigma^2}\right)$, where $\|.\|$ represents the Euclidean norm, and σ is a bandwidth parameter.

MLP Multilayer Perceptron: $f(\hat{\mathbf{y}}) = \sum_{j=1}^{m} v_j \phi_j(\hat{\mathbf{y}}) + v_0$, $\phi_j(\hat{\mathbf{y}}) = \frac{2}{1+\exp\left(-\left(\sum_{i=1}^{n} w_{i,j} \hat{y}_i + w_{0,j}\right)\right)} - 1$, where m denotes the number of hidden nodes, and w and v are the weights of the hidden and output layers, respectively.

RF Random Forest: $f(\hat{\mathbf{y}}) = \sum_{\tau \in \Xi} w_\tau(\hat{\mathbf{y}}) y_\tau$, $w_\tau(\hat{\mathbf{y}}) = \frac{1}{p} \sum_{j=1}^{p} \frac{\mathbb{1}\{\hat{\mathbf{y}}_\tau \in \ell_j(\hat{\mathbf{y}})\}}{\sum_{\kappa \in \Xi} \mathbb{1}\{\hat{\mathbf{y}}_\kappa \in \ell_j(\hat{\mathbf{y}})\}}$, where p is the number of trees in the forest, ℓ_j denotes the leaf that is obtained when dropping $\hat{\mathbf{y}}$ down the j-th tree, and $\mathbb{1}$ denotes the indicator function.

LSTM Long Short-Term Memory: $f(\hat{\mathbf{y}}_t) = \mathbf{v}^T \mathbf{h}_t(\hat{\mathbf{y}}_t) + v_0$, where $\mathbf{h}_t(\hat{\mathbf{y}}_t) = \text{LSTM}(\hat{\mathbf{y}}_t, \mathbf{c}_{t-1}, \mathbf{h}_{t-1}; \mathbf{w}) \in \mathbb{R}^m$, $\mathbf{w}$ and $\mathbf{v}$ are the weights of the LSTM and linear layers, respectively, and $\mathbf{c}$ and $\mathbf{h}$ are the cell and hidden states.

3.2. Probabilistic Forecasting

QRS Quantile Estimation through Residual Simulation: This method operates on the premise that the distribution of residuals observed in historical data remains consistent with the distribution for future data. The process entails the following steps: (i) training a meta-model for deterministic forecasting,

(ii) computing residuals for historical data, (iii) adding these residuals to the point forecasts, and fitting a distribution function to the resultant values using a nonparametric kernel method, and (iv) calculating quantiles using the inverse cumulative distribution function (CDF).

QLR Quantile Linear Regression: The linear model in the form of LR can effectively generate quantiles when it minimises the pinball loss function [5]: $L_\alpha(y, q) = (y - q)(\alpha - \mathbb{1}\{y < q\})$, where y represents the true value, and q is its predicted α-quantile.

QRF Quantile Regression Forest: QRF focuses on estimating the full CDF, which is approximated through the weighted mean [6]: $\hat{F}(y|X = \hat{\mathbf{y}}) = \sum_{\tau \in \Xi} w_\tau(\hat{\mathbf{y}})$ $\mathbb{1}\{y_\tau \leq y\}$, where $w_\tau(\hat{\mathbf{y}})$ are determined in the same way as for RF. Based on the empirical CDF, the conditional quantiles are calculated.

4. Experiments

We assess our stacking methods for short-term electrical load forecasting (STLF) across 35 European countries using hourly data from 2006 to 2018 [7]. We employed a diverse array of forecasting base models, including statistical and ML models (recurrent, deep, and hybrid NN architectures) sourced from [7]. These models generated hourly forecasts for 2018. We then selected 100 specific hours for each country from the latter half of 2018 (evenly distributed). The forecasts for these selected test hours were aggregated by meta-models. Training data span from January 1, 2018, to the hour before the forecasted hour ($h = 1$). The meta-models were optimized for hyperparameters and then trained using both global and local modes. In the local modes, we employed $k \in \{20, 40, ..., 200, 250, 300\}$. For LSTM, we assumed $s_1 = 24$, $s_2 = 168$, and $c \in \{24, 48, 72, 168, 504\}$. For probabilistic STLF, we specified $\Pi = \{0.01, 0.02, ..., 0.99\}$.

Table 1 provides a summary of the results for deterministic forecasting, showcasing quality metrics for the top-performing variants of the meta-models, alongside Mean and Median benchmarks for comparison. Mean and Median represent the average and median of the 16 forecasts generated by the base models, respectively.

An examination of the table reveals RF's outstanding performance in terms of MAPE and MdAPE, while LSTM achieves the lowest MSE. Additionally, kNN and MLP demonstrate competitive accuracy. LR exhibits lower accuracy but it excels in forecast bias, as indicated by MPE. Importantly, all proposed meta-learners

outperform the Median and Mean methods in both forecast accuracy and dispersion, as measured by StdPE.

Table 1. Quality metrics for point forecasting.

Meta-model	MAPE	MdAPE	MSE	MPE	StdPE
Mean	1.91	1.23	316943	–0.0775	3.11
Median	1.82	1.13	287284	–0.0682	3.05
LR, global	1.63	1.11	213428	**0.0131**	2.38
kNN, $k = 40$	1.54	1.03	178699	–0.0915	2.33
MLP, $k = 120$	1.59	1.09	180839	–0.0786	2.31
RF, global	**1.52**	**1.01**	173821	–0.0837	**2.26**
LSTM, v1, $c = 168$	1.55	1.09	**139667**	0.0247	**2.26**

MAPE – mean absolute percentage error, MdAPE – median of APE, MSE –mean squared error, MPE – mean PE, StdPE – standard deviation of PE.

Table 2 provides a concise summary of the quality metrics for probabilistic forecasting. It reveals that QRF emerges as the top-performing meta-model. It demonstrates the lowest MPQRE, MARFE, MPWS and QMAPE. Both QRF and QLR exhibit similar PIs, which closely resemble the ideal distribution of 90% PI width and 5% coverage on each side. In contrast, the PIs generated by QRS are significantly narrower, covering, on average, less than 60% of the target values. When considering QMAPE calculated based on the medians of the predicted distributions, it becomes evident that these values are larger than those generated by the meta-models designed for point forecasting (as seen in Table 1).

Table 2. Quality metrics for probabilistic forecasting.

Meta-model	MPQRE	MARFE	MPWS	inPI/belowPI/abovePI	QMAPE
QRS, global	0.735	0.1163	18.38	59.26/18.40/22.34	1.78
QLR, global	0.669	0.0327	12.62	**90.34**/**4.74**/4.91	**1.75**
QRF, global	**0.657**	**0.0321**	**11.64**	90.77/4.23/**5.00**	**1.75**

MPQRE – mean percentage quantile regression error ($MPQRE = \frac{100}{N|\Pi|}\sum_{i=1}^{N}\sum_{\alpha\in\Pi}\frac{L_\alpha(y_i,\tilde{q}_i(\alpha))}{y_i}$, where y_i represents an observed value), MARFE – mean absolute relative frequency error ($MARFE = \frac{1}{|\Pi|}\sum_{\alpha\in\Pi}|ReFr(\alpha)-\alpha|$, $ReFr(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\{y_i \le \tilde{q}_i(\alpha)\}$), MPWS – mean percentage Winkler score for 90%-PI ($MPWS = \frac{100}{N}\sum_{i=1}^{N}\frac{WS(y_i,\tilde{q}_{l,i},\tilde{q}_{u,i})}{y_i}$, where WS is the Winkler score, $\tilde{q}_l$ and $\tilde{q}_u$ are the predicted lower and upper quantiles defining the 90%-PI), inPI/belowPI/abovePI – percentages of observed values in 90%-PI, below PI and above PI (desired values are 90/5/5), QMAPE – MAPE for assessing point forecasts calculated based on predicted quantiles; it assumes that the point forecast corresponds to the 0.5-quantile (median).

5. Conclusion

In this study, we consistently found that meta-models surpassed traditional forecast combination methods like Mean and Median for point STLF across most scenarios. Notably, nonlinear models showed superior accuracy over LR. Among them, RF emerged as the top performer, offering the most accurate predictions.

In the realm of probabilistic forecasting, we introduced three methods for generating quantile forecasts based on base point forecasts. Notably, QRF outperformed QLR and showed significant improvement over a method relying on residual simulation. Global learning outperformed in probabilistic STLF, whereas local learning may offer advantages in specific scenarios for deterministic STLF.

References

[1] Wang, X., Hyndman, R. J., Li, F., and Kang, Y. Forecast combinations: An over 50-year review. *Int. J. Forecast.*, 39(4):1518–1547, 2023.

[2] Babikir, A. and Mwambi, H. Evaluating the combined forecasts of the dynamic factor model and the artificial neural network model using linear and nonlinear combining methods. *Empir. Econ.*, 51(4):1541–1556, 2016.

[3] Gastinger, J., Nicolas, S., Stepić, D., Schmidt, M., and Schülke, A. A study on ensemble learning for time series forecasting and the need for meta-learning. In *Proc. 2021 IEEE IJCNN*, pages 1–8. 2021.

[4] Dudek, G. Combining forecasts using meta-learning: A comparative study for complex seasonality. In *Proc. 2023 IEEE DSAA*, pages 1–10. 2023.

[5] Koenker, R. and Hallock, K. F. Quantile regression. *J. Econ. Perspect.*, 15(4):143–156, 2001.

[6] Meinshausen, N. Quantile regression forests. *J. Mach. Learn. Res.*, 7:983–999, 2006.

[7] Smyl, S., Dudek, G., and Pełka, P. ES-dRNN: A hybrid exponential smoothing and dilated recurrent neural network model for short-term load forecasting. *IEEE Trans. Neural Netw. Learn. Syst.*, pages 1–13, 2023.

Chapter 2

Knowledge Engineering

Track Chairs:

- prof. Agnieszka Ławrynowicz - Poznan University of Technology
- prof. Dariusz Krol - Wrocław University of Science and Technology
- prof. Grzegorz J. Nalepa - Jagiellonian University

Generating TREC Compatibile Clinical Trials Knowledge Graph for Information Retrieval

Jakub Dutkiewicz[0000−0002−7954−7484],
Czesław Jędrzejek[0000−0002−9593−6818]

Poznan University of Technology
Marii Skłodowskiej-Curie 5, 60-965 Poznan, Poland
jakub.dutkiewicz@put.poznan.pl
czeslaw.jedrzejek@put.poznan.pl

Abstract. *This work is motivated by the analysis of TREC PM Topic 5 official evaluation results, where we found a discrepancy compared with our human evaluation. To automate the evaluation process, we introduce a clinical trials knowledge model. We utilize resources that define very specific medical concepts included in the TREC PM queries, namely disease, gene, and treatment, each with its concept relevance. One focal point of each clinical trial, as well as each TREC PM query, is the disease for which we augment the Disease Ontology. We design a relevance formula containing weights accounting for ontological relatedness. Additionally, we employ the GPT model to identify lists of drugs related to specific genes. Our results demonstrate significant agreement in the "No Match" (not relevant) and "Exact" categories, but substantial disagreement in partial match categories, and very large disagreement in the "Not PM" category.*
Keywords: *knowledge extraction, knowledge modelling*

1. Introduction

Information Retrieval(IR) deals with searching and evaluating information from document repositories. Given a set of documents and a list of queries, the task is to return a list of documents sorted by the relevance of each document to the given query. TREC-PM[1][2] was (and is in a TREC-CT version) a biomedical track, which evaluates the systems performing the IR tasks on a specific set of documents and clinical trials and queries called topics. The TREC PM in 2017-2019

focused on the field of oncology. The topics are divided into several fields: diseases, related genes; each topic also contains demographic descriptions of patients, including their age and gender. In this work we are concerned with clinical trials, so each query defines a potential patient in a cohort, that matches the given trial requirements. An analysis of clinical trial documents faces significant difficulty due to several reasons:

1. The clinical trial documents were not planned to have semantic meaning. They are partially formalized by the element structure. Unfortunately, there are cases in which the formal structure is not followed to the letter.

2. As in the previous statement information on a given feature is distributed across many elements and importance for relevancy is not equal between elements.

Assessors are supposed to evaluate the trials in the scope of:

1. Relationship between the clinical trial and the disease in the topic. Here, the annotators have four options: (1) an exact match, (2) a more specific disease, (3) a more general disease, and (4) a different disease.

2. Determining the significance of the clinical trial in the context of genes included in the topic. Every gene is evaluated separately. There are four possible annotations here: (1) exact match, (2) the trial is in a way related to this gene, but to a variant, (3) there is no information on this gene in the trial, and (4) the trial is related to this gene, but there is no information on the variant of the gene in the trial.

3. Patient demographic information, such as age and gender, and other features.

These are the general TREC annotation guidelines. The specific description of the annotation process can be found in the original publication [2].
We conceived the idea for this work during our human evaluation of the TREC PM 2017 Topic 5 data, where we uncovered a significant discrepancy between our results and the TREC evaluation. The rules for determining relevance for a given document require further elaboration compared to the original TREC PM 2017 guidelines.

2. Domain level ontology

This work encompasses two levels of medical knowledge abstraction. The upper tier of the knowledge model comprises domain-specific resources. We necessitate an extensive linguistic representation set for each of the utilized concepts, as these resources will be employed in an automated entity linking process - connecting the textual representation of a trial to an abstract knowledge concept. We model three types of domain-level concepts, with particular attention to the practicality of the vocabulary used to describe them:

1. Diseases - in this case, we utilize an external middle-level ontology - the Disease Ontology (DO). After initial inspection, we decided to slightly expand the vocabulary using the *hasExactSynonym* relation[1].
2. Genes - we propose a handcrafted set of gene synonyms[1].
3. Drugs - we employ the GPT language model to create a preliminary list of drugs targeting chosen genes. The list is then carefully checked and pruned of nonsensical information.

The lower tier of knowledge covers the information on Clinical Trials and TREC PM queries. Each query is represented as an entity of **TREC Query** class, with several data annotation properties, such as Topic Number, Year and description of Demographics. Each Query is linked to a disease and a gene via a **QueryLink**. On the other hand, each document is represented by a **Clinical Trial** entity. The clinical trial *consists of* **Clinical Trial Fields**, such as **Brief Title**, **Description**, **Criteria** and others. The lower tier of knowledge is interconnected with the upper tier through entity links. The links are discovered using dedicated string mappers and a tool called LexMapr (Lexical Mapper) [3]. We can employ this approach thanks to the vast vocabulary in the upper tier and the appropriate granularity of knowledge in the lower tier. A full knowledge model is depicted in Fig. 1.

3. Knowledge Graph

In this section, we present a small sample of the knowledge base, representing one of the documents relevant to the 5th query from TREC PM 2017. The

[1] publicly available on www.github.com/dudenzz/TRECCTKB

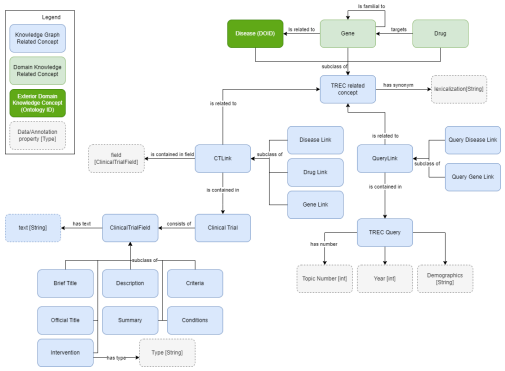


Figure 1. TREC compatible Clinical Trials Knowledge Model.

query specifications are as follows: *Disease:* **Melanoma**, *Genes:* **BRAF(V600E)**, **CDKN2A Deletion**, *Demographics:* **45-year-old female**. The evaluated Clinical Trial has the id: **NCT01166126**. In fact it is a document relevant to the given query. The title *Temsirolimus/AZD 6244 for Treatment-naive With BRAF Mutant Unresectable Stage IV* indicates that it is related to the BRAF gene, which is captured by the **Gene Link** discovered for both the brief and official title of the trial. The indication of a disease is mentioned in the **Condition** field, which is also discovered by the Entity Linker. We present the described sample in the Fig. 2.

4. Retrieval with Knowledge Graph

In this section, we present a retrieval score function, which is calculated upon the knowledge graph. We focus on evaluating the gene relevance score. It should be noted that several genes in the upper-tier knowledge base are annotated with a **familial gene** relation. It is a relation between genes, which indicates genes belonging to the same family. The score is calculated based on the number of genes linked to a document. Exact matches (where the entity links match the gene provided in the query) are annotated by *ex* in the following equation, representing the number of gene links of this specific type within the document. Genes belonging to the same family as the query gene (excluding exact matches), are annotated by *dv*. Linked genes that are completely different from the query genes are denoted by *nm*. Subscripts in the equation below relate to the Clinical Trial fields where

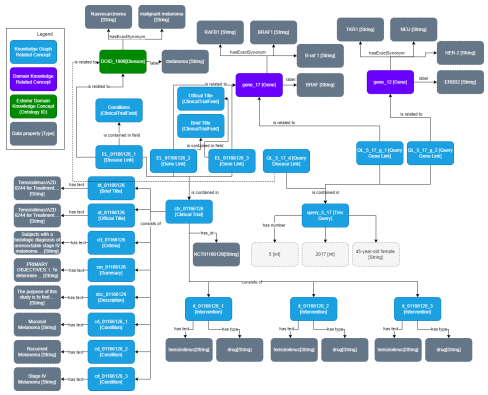


Figure 2. A sample of a Knowledge Base

the genes were found. *bt* and *ot* relate to **official** and **brief title** respectively, while *summ*, *desc* and *crit* relate to **summary**, **description** and **criteria** fields respectively. Finally, the *drug* annotation relates to the number of Drug Links targeting one of the genes from the query. w_1, w_2 and b are hyperparameters of the model. The retrieval model for the gene relevance score is defined as follows:

$$score_{gene} = \begin{cases} \sqrt[drug+1]{\frac{(ex_{bt}+ex_{ot})+0.5\cdot(dv_{bt}+dv_{ot})}{ex_{bt}+ex_{ot}+dv_{bt}+dv_{ot}+nm_{bt}+nm_{ot}}} & ex_{bt}+ex_{ot}+dv_{bt}+dv_{ot}>0 \\ \sqrt[drug+1]{\frac{1}{1+e^{-w_1\cdot(w_2\cdot(dv_{crit}+dv_{summ}+dv_{desc})+ex_{crit}+ex_{summ}+ex_{desc})-b}}} & otherwise \end{cases} \quad (1)$$

This model focuses on the significance of the title. We hypothesize that the title of a trial is the most informative summary of the entire trial. If a specific gene appears within the title, it is a strong indicator that the trial is related to the given gene. However, if the gene does not appear in the title, we employ a sigmoid-shaped retrieval model function. A specific number of gene mentions in other parts of the trial "triggers" the function, causing the model to return a high value. The score is further boosted if a mention of a gene-related drug is apparent in the text. Using this model, we measure the compatibility of our approach with the TREC PM annotations. The aggregated average result of 78% compatibility is a good prognosis. The compatibility results are presented in Tab. 1.

	Relevant	Partially relevant	Non relevant
Exact	3773	1169	450
Different Variant	802	135	179
Missing Variant	1551	624	411
No match	671	866	4858

Table 1. Compatibility of with TREC PM annotations

5. Conclusions

In this work, we present a knowledge model consisting of concepts pertinent to clinical trial documents. We are able to instantiate this model with clinical trial documents and TREC PM queries. Matching between these two sources of data forms the basis for relevance calculation. So far, gene matching has been achieved. Compared with the official evaluation for TREC PM 2017-2019 (cancer-related queries), our results show quite an agreement in the No Match (not relevant) and Exact categories, and significant disagreement in partial match categories. A full analysis will require handling the NotPM category, which also depends on genes. We claim that the use of the NotPM category saves some evaluation time but also obscures evaluation.

References

[1] Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., and Lazar, A. J. Overview of the TREC 2018 precision medicine track. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018*. 2018.

[2] Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., Lazar, A. J., and Pant, S. Overview of the TREC 2017 precision medicine track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC*. 2017.

[3] Gosal G, D. D. e. a., Griffiths E. Lexmapr: a rule-based text mining tool for ontology-driven harmonization of short biomedical specimen descriptions. 2019.

Chapter 3

Medical Applications of Artificial Intelligence

Track Chairs:

- prof. Włodzisław Duch - Nicolaus Copernicus University in Toruń
- prof. Julian Szymański - Gdańsk University of Technology, Gdańsk

Automatic detection of contraction direction in cardiomyocytes

Maciej Szymkowski[1,2][0000−0002−5778−9976], **Jakub Gołąb**[1][0009−0000−5336−9141], **Kacper Perz**[1][0009−0004−5733−2820], **Bartosz Poniewierka**[1][0009−0007−9743−1374]

[1]*Łukasiewicz Research Network - Poznań Institute of Technology*
Computer Science Research Group, Division of Artificial Intelligence Development
Ewarysta Estkowskiego 6, 61-755, Poznań, Poland
{maciej.szymkowski, jakub.golab, kacper.perz, bartosz.poniewierka}@pit.lukasiewicz.gov.pl
[2]*Białystok University of Technology*
Faculty of Computer Science
Wiejska 45A, 15-351 Bialystok, Poland
m.szymkowski@pb.edu.pl

Abstract. *One of the important problems in cardiac experiments is connected with high risk of heart stoppage or myocardial infarction. This is the main reason why such processes should be performed in a simulated, save environment. The goal of our research is to propose a tool that will enable observation of contractions within single cardiomyocyte. Experienced cardiologists claimed that it is of high importance to determine contractions and their directions – especially after application of new medication to the cell. The main goal of our work is to observe a single Engineered Heart Tissue (EHT). with the high-resolution camera and then to determine when contraction is observed and what its direction is. To fulfil that aim we consume recent ideas – like OpticalFlow, but also, we propose our own additional stages. It needs to be claimed that the results of the analysis clearly point out that it is possible with high precision and accuracy to observe contractions and evaluate their direction.*

Keywords: *Cardiomyocytes, 3D models, heart, contraction direction, optical flow, Engineered Heart Tissues (EHTs)*

DOI:10.17388/WUT.2024.0002.MiNI

1. Introduction

Ability to contract is one of the most principal properties of cardiomyocytes. However, under some circumstances not all contractions happen in the same manner of force and direction, this situation can be observed as a heart failure [2]. It is especially visible in the process of differentiating between subpopulations of iPSC derived cardiomyocytes - their introduction to hearth in unseparated manner can result in life threatening arrhythmia. [1]

In order to clearly separate cells with different properties of contraction each cell have to be observed during multiple contractions cycles. Afterwards motion vector analysis should be performed in order to obtain maximum motion displacement.

There are diversified algorithms performing such computations but those that are the most promising consume methodology called OpticalFlow and have numerous advantages as ability to detect small displacements or low computational costs. In this work, the authors would like to present an approach to detection of the contraction direction in artificially grown cardiomyocytes. In our work, OpticalFlow is also consumed. It needs to be pointed out that the experiments were performed with the dataset collected by the Team of prof. Tomasz Kolanowski from Institute of Human Genetics, Polish Academy of Science. The database consists of 8 movies presenting single cardiomyocyte in the contraction cycle (the duration of each sample is no longer than 25 seconds).

The work is divided into several subchapters. The first of them concerns short introduction to the main area of the work. In the second one, we present information about the recent state of the art (from computational point of view). The third subchapter presents the proposed approach (with sufficient details) whilst in the last one, conclusions and future work are given.

2. Recent state of the art

OpticalFlow is one of the most useful approaches in the field of cardiomyocytes movements analysis. Several different approaches consume it for detection of contractions in EHTs. However, many researchers have used variations of block matching algorithms where each frame is divided into squares $N \times N$. Then in the next frame the same square is found and the difference in position is calculated as perceived motion [6]. This method is considered as reliable in coupling between

calcium flux and detectable heart beat rate (BPM) [10].

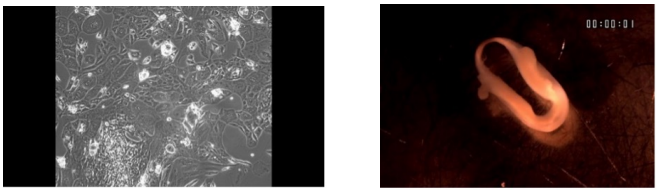
Others have used the approach proposed by Farnebäck [7]. Algorithm based on polynomial expansion has proven capable in detecting drug induced response in contractility [8]. Additionally, its usefulness in determining differences in beat rates (BPM) between IPSC's cardiomyocytes derived from two separate protocols was claimed. Moreover, it has detected disparities in clusters originating from the same protocol resulting in a more precise way to access maturation process of cardiomyocytes. [9]. On the other hand, we can also find diversified approaches. For example, in work [11] the Authors proposed an algorithm where points in the cell are selected randomly – however, they are stable during the whole process and are used for tracking. Interesting idea was also presented in work [12]. The Authors also consumed OpticalFlow for observation of tissue movement, however, in the second part of their experiments they used the results from OpticalFlow to observe the central point of the contraction. It was done by analysis of data convergence.

3. Proposed approach and results of the experiments

The Authors would like to propose slightly different approach than it is described in recent publications. We will reuse the OpticalFlow library, however, within completely different samples. Until now, most of the works concern 2D images. We present them in Fig 1a. However, we would like to observe the cells that are presented in 3D (it means that we will not consume microscopic images but rather we will have a broad view of the cell) – the representative sample is showed within Fig. 1b. In the worked-out approach we are performing two general stages. The first is observation of the 4 points in the cell. These are the points that are placed at the far ends of the cell (in top, bottom, left and right). To obtain the most precise result, at the very beginning we segment the cell from the background with remBG algorithm [13]. After that we perform mathematical modelling to obtain information about previously mentioned 4 points in the cell. It means that we determine the center of the cell (with ready to use methods from OpenCV library) and then we calculate the parameters of two perpendicular lines passing through the center – the places where the lines meets the perimeter of the cell allows us to determine 4 points used for further observation of the cell movement.

When all these points are calculated, we track them with usage of OpticalFlow algorithm. The Lucas-Kanade method is used to obtain information about the

movement of the points. By that idea we are able to determine contractions and their directions. It allows us to observe whether the contraction is done appropriately (all points should move to inside or outside of the cell simultaneously – systolic or diastolic stage). The results of the proposed algorithm are given in Fig. 2. It shows the movements of all selected points (both in horizontal direction – "x", and vertical direction – "y"). In Fig. 3, we present the scheme of the solution.



(a) (b)

Figure 1: The view of 2D microscopic sample (a) and 3D model of the heart cell (b).

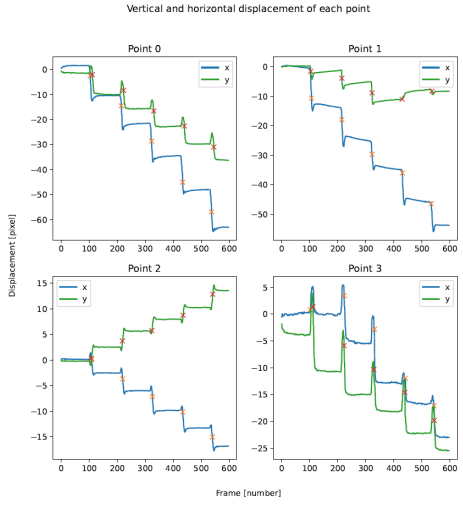


Figure 2: Charts presenting movement of the 4 points with marked moments of contraction

4. Conclusions and Future Work

The goal of the work was to introduce an approach for observation of contractions and their directions in 3D cardiomyocytes samples. The novelty of that work lies in the usage of combination of computational methods (OpticalFlow) and Artificial Intelligence-based algorithms (like remBG) to determine the parameters mentioned above in EHTs (Engineered Heart Tissues). We observed that contraction directions are determined correctly. It was also proven by experienced cardiologists. It was claimed that the proposed algorithm can be a part of a real medical software that can be used by medical technicians and cardiologists during experiments. Moreover, it can be useful in the case of new medications testing.

The next steps in our work will be related to enlarge the database (at least 100 movies) as well as to compare the worked-out solution with those used in real medical processes. It will be also crucial to develop an algorithm for detection of parameters like contraction direction or area of contraction. Such an algorithm can also be used as a benchmark for further experiments.

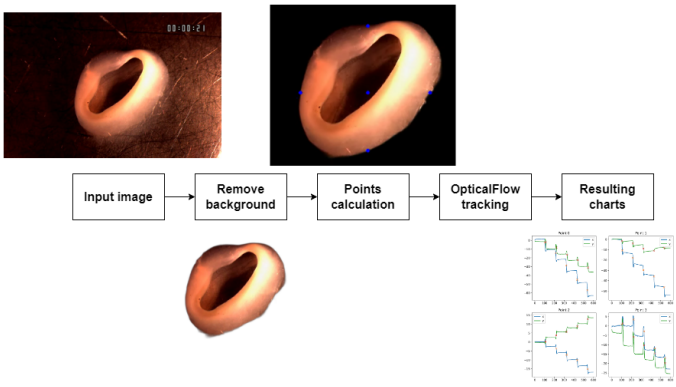


Figure 3: Scheme of the proposed procedure and all its stages

Acknowledgment

This work was partially supported by grant WI/WI-IIT/4/2022 from Białystok University of Technology and partially by the funds of Łukasiewicz Research Net-

work – Poznań Institute of Technology and funded with resources for research by Ministry of Science and Higher Education in Poland.

References

[1] Kolanowski, T. J., Antos, C. L., & Guan, K. (2017). Making human cardiomyocytes up to date: Derivation, maturation state and perspectives. International journal of cardiology, 241, 379-386.

[2] Mora, M. T., Zaza, A., & Trenor, B. (2023). Insights from an electromechanical heart failure cell model: Role of SERCA enhancement on arrhythmogenesis and myocyte contraction. Computer Methods and Programs in Biomedicine, 230, 107350.

[3] Sugiyama, A., Hagiwara-Nagasawa, M., Kambayashi, R., Goto, A., Chiba, K., Naito, A. T., ... & Izumi-Nakaseko, H. (2019). Analysis of electromechanical relationship in human iPS cell-derived cardiomyocytes sheets under proarrhythmic condition assessed by simultaneous field potential and motion vector recordings. Journal of Pharmacological Sciences, 140(4), 317-320.

[4] Eisner, D. A., Caldwell, J. L., Kistamás, K., & Trafford, A. W. (2017). Calcium and excitation-contraction coupling in the heart. Circulation research, 121(2), 181-195.

[5] Rast, G., Kraushaar, U., Buckenmaier, S., Ittrich, C., & Guth, B. D. (2016). Influence of field potential duration on spontaneous beating rate of human induced pluripotent stem cell-derived cardiomyocytes: Implications for data analysis and test system selection. Journal of Pharmacological and Toxicological Methods, 82, 74-82.

[6] Hayakawa, T., Kunihiro, T., Dowaki, S., Uno, H., Matsui, E., Uchida, M., et. al. (2012). Noninvasive evaluation of contractile behavior of cardiomyocyte monolayers based on motion vector analysis. Tissue Engineering Part C: Methods, 18(1), 21-32.

[7] Farnebäck, G. (2002). Polynomial expansion for orientation and motion estimation (Doctoral dissertation, Linköping University Electronic Press).

[8] Ahamadzadeh, E., Jaferzadeh, K., Park, S., Son, S., & Moon, I. (2022). Automated analysis of human cardiomyocytes dynamics with holographic image-based tracking for cardiotoxicity screening. Biosensors and Bioelectronics, 195, 113570.

[9] Nitsch, S., Braun, F., Ritter, S., Scholz, M., & Schroeder, I. S. (2018). Functional video-based analysis of 3D cardiac structures generated from human embryonic stem cells. Stem cell research, 29, 115-124.

[10] Huebsch, N., Loskill, P., Mandegar, M. A., Marks, N. C., Sheehan, A. S., Ma, Z., et al. (2015). Automated video-based analysis of contractility and calcium flux in human-induced pluripotent stem cell-derived cardiomyocytes cultured over different spatial scales. Tissue Engineering Part C: Methods, 21(5), 467-479.

[11] Zhang, Y., Zhao, X., Liu, Y (2023). A visual detection method of cardiomyocyte relaxation and contraction. AIP Advances, 13, art. no. 025028, DOI: 10.1063/5.0133456

[12] Czirok, A., Isai, D., Kosa, E., et. al. (2017). Optical-flow based non-invasive analysis of cardiomyocyte contractility. Scientific Reports, vol. 7, art. no. 10404, DOI: 10.1038/s41598-017-10094-7

[13] RemBG segmentation model - https://github.com/danielgatis/rembg (Accessed 02.02.2024)

A comparative analysis of deep learning models for lung segmentation on X-ray images

Weronika Hryniewska-Guzik[0000−0003−2903−6050], **Jakub Bilski**[*], **Bartosz Chrostowski**[*], **Jakub Drak Sbahi**[*], **Przemysław Biecek**[0000−0001−8423−1823]

Warsaw University of Technology
Faculty of Mathematics and Information Science
Koszykowa 75, 00-662 Warsaw, Poland
weronika.hryniewska.dokt@pw.edu.pl

Abstract. *Robust and highly accurate lung segmentation in X-rays is crucial in medical imaging. This study evaluates deep learning solutions for this task, ranking existing methods and analyzing their performance under diverse image modifications. Out of 61 analyzed papers, only nine offered implementation or pre-trained models, enabling assessment of three prominent methods: Lung VAE, TransResUNet, and CE-Net. The analysis revealed that CE-Net performs best, demonstrating the highest values in dice similarity coefficient and intersection over union metric.*
Keywords: *Semantic segmentation, X-ray, Lungs, Deep Learning, U-Net*

1. Introduction

In the field of medical imaging, accurate segmentation of lungs on X-rays is important in many applications [1], from early disease detection to treatment planning and patient monitoring. As healthcare evolves, the need for fast and accurate tools grows, implying physician support with deep learning approaches [2]. In particular, solutions such as U-Net demonstrate the potential to automate the task of lung segmentation, offering promising advances in improved accuracy [3].

[*]equal contribution

DOI:10.17388/WUT.2024.0002.MiNI

However, despite these advances, the inevitable diversity of X-ray images makes it difficult for some modern segmentation methods to deal with them. Although many solutions show high performance in simple and typical cases, their performance degrades when confronted with complex ones. Moreover, the issue of using pre-trained models on images with different characteristics may have potential negative consequences when used for real-world solutions [4].

Recognizing these challenges, our objective is to analyze existing solutions for lung segmentation and systematically evaluate their performance across a dataset of varying characteristics. In this study, we analyze and compare three prominent methods - Lung VAE, TransResUNet, and CE-Net - using five image modifications. The ultimate goal is to determine the most accurate method for lung segmentation in diverse scenarios.

2. Methodology

The complexity of the lung segmentation task is related to the scarcity of actual data containing X-ray images and its masks with diverse obstructions such as jewelry, advanced stages of disease, and some medical devices present in a patient's body. To address this limitation, we merged two existing datasets: Montgomery County X-ray [5] and Shenzhen Hospital X-ray [6]. The first one contains 138 X-rays, of which 80 are normal and 58 are abnormal with manifestations of tuberculosis. Alongside the lung, segmentation is provided. The second one contains 340 normal and 275 abnormal X-rays showing various manifestations of tuberculosis. The masks are provided by Stirenko et al. [7].

For our analysis of models dedicated for lung field segmentation, we select 54 methods documented in Çallı et al. [8] and seven scientific papers which implementation is available on Github platform models. Most of the solutions could not be run due to obsolete versions of libraries, lack of the reproduction steps in the article/repository, not working parts of code, and lack of methods for using data other than those provided with code. Finally, we are able to run only three of them, described in Section 3.

If there was a pre-trained model available for any of the architectures, it was used for the evaluation process. Otherwise, a model was trained on the data on which it was originally trained, as it was described in the source article.

Then, an evaluation was performed on the prepared test data. It contained both original and augmented images. For every pair of ground-true and predicted

mask, the quality of segmentation was assessed using dice similarity coefficient [9] and intersection over union (IoU). The following augmentations were done: contrast, random rotation, bias field, horizontal flip, and discrete "ghost" artifacts. The resulting dataset allowed us to perform testing on model behavior when presented with an image with augmentation that was not present in the training phase.

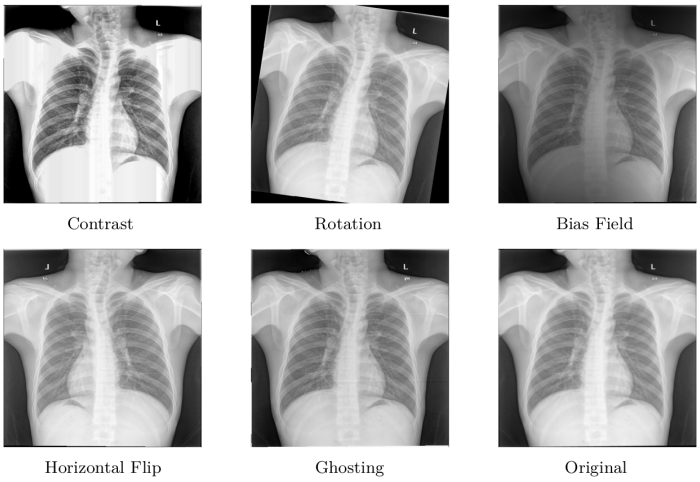


Figure 1: Types of augmentation on which the ability to adapt to new conditions of segmentation models was tested.

3. Related Works

Lung VAE [10][1] aim is to be a lung segmentation model that is not dependent on training set distribution. For this purpose, a U-net-based segmentation network encoder and variational encoder is used. As the authors did not train the model on opacity cases, the variational encoder is used for data imputation and may be treated as data augmentation. The results of U-net encoder and variational encoder are concatenated and passed to the decoder.

TransResUNet [11][2] is an improved version of U-net [3] model dedicated for the task of lung field segmentation. Three modifications were made. First,

[1] https://github.com/raghavian/lungVAE

[2] https://github.com/sakibreza/TransResUNet

a pre-trained encoder was taken from VGG-16 architecture trained on the ImageNet dataset. The first seven layers were used. As a second improvement, between the encoder and decoder layers were added skip connections with a series of convolution blocks. Finally, a dedicated post-processing step with hole filling, artefacts removal and morphological opening was applied to the output images.

CE-Net [12][3] is n U-net-based approach of a context encoder network. Its aim is to capture more abstract and preserve spatial information for 2D medical image segmentation. There are three components: a feature encoder module, a context extractor, and a feature decoder module. For feature extraction, pretrained ResNet-34 is used. Dense atrous convolution block and residual multi-kernel pooling are used for context extraction. In feature decoder module, transposed convolution is applied.

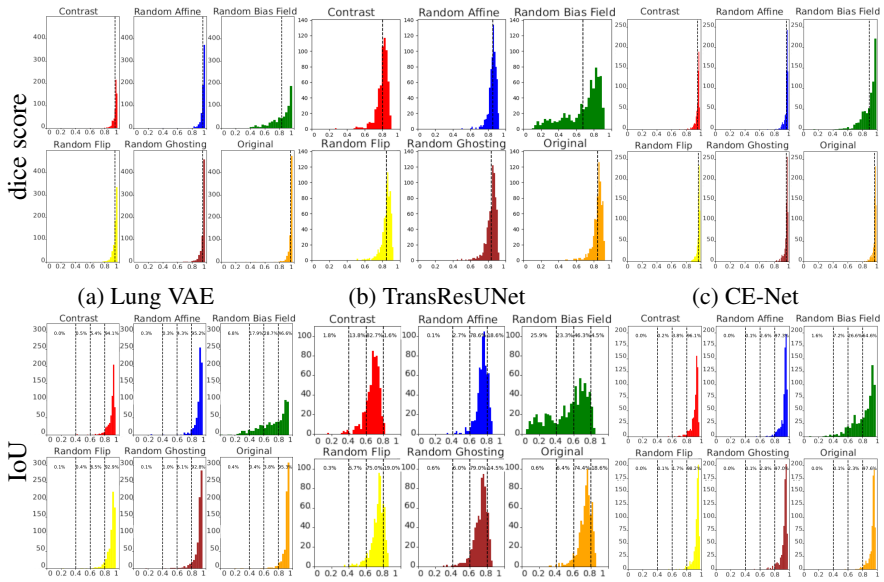
4. Results and discussion

The Lung VAE model, shown in Figure 2a, obtained the best results for images without augmentations and the worst for the Random Bias Field augmentation. From the augmented images, it seems to have the best results for the Random Ghosting.

Figure 2b illustrates the results for the TransResUNet model, which performed less favorably compared to the other two methods. The mean dice loss for images without augmentation was slightly greater than 80%. Three augmentation types had no real effect on the results; Random Affine, Random Flip, and Random Ghosting achieved a very similar mean dice loss as the case with no augmentation. The two remaining augmentation methods posed a more serious problem. Contrast got a slightly lower dice loss and a thicker tail of the distribution, while Random Bias Field performed much worse, with a very long tail and low mean score. While other methods also struggled with this type of augmentation, this method seemed to perform the worst.

The results for CE-Net are presented in Figure 2c, showcasing its outperformance relative to both Lung VAE and TransResUNet. This distinction is particularly noticeable for the Random Bias Field augmentation, where CE-Net exhibited a better average dice similarity coefficient and a significantly shorter low-tail compared to the other two models.

[3] `https://github.com/Guzaiwang/CE-Net`



(a) Lung VAE (b) TransResUNet (c) CE-Net

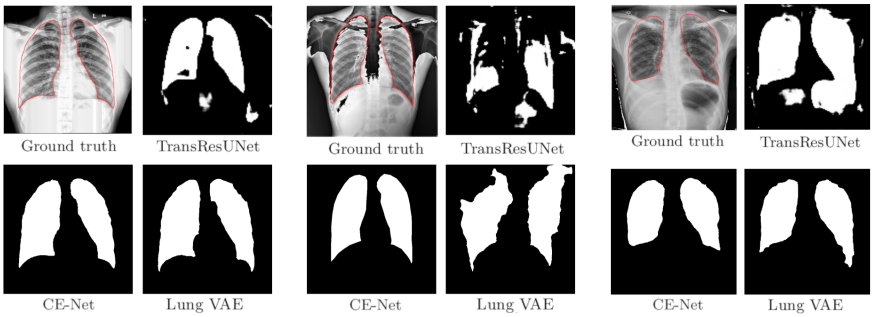
(d) Lung VAE (e) TransResUNet (f) CE-Net

Figure 2: Segmentation results (dice similarity coefficient and IoU value) after applying various augmentation methods that have not been performed on the training set before. On the horizontal axis is the dice similarity coefficient value from 0 to 1, and on the vertical axis is the number of samples from 0 to 500.

Figure 2d, 2e, and 2f illustrate the results for the IoU score, providing a summary of the score distribution. Here, the robustness of CE-Net is evident, with very few samples falling outside the best score bin. In contrast, TransResUNet displayed a thicker tail, with some samples consistently reaching the worst bin and a peak in the third bin. Although Lung VAE performed slightly worse than CE-Net in all cases, it still achieved significantly better results than TransResUNet, emphasizing CE-Net as the superior and more reliable algorithm.

Figure 3 shows an example of the segmentation results of the original image for three models using different previously unseen augmentations. As was presented in the aggregated analysis, the CE-Net model in all augmentations managed to obtain satisfying results. The most significant differences between outputs of Ce-Net and Lung VAE are visible in Figure 3b. The Lung VAE model captured

the position of the lung. However, the obtained mask did not manage to preserve the shape, especially in the top part of the image. TransResUNet often detects spots outside of the lung that seem to correspond with places with a darker shade of gray than the rest of the body.



(a) contrast (b) bias field (c) random ghosting

Figure 3: Original X-ray images with segmented lungs (marked in red) compared to lung masks generated by various models using different augmentation methods.

5. Conclusions

In literature, the vast majority of solutions for lung segmentation are based on the U-net architecture enhanced with either the preprocessing of data or the post-processing of output masks. It is worth stressing that when proposing a new architecture, it is necessary to ensure that the results are reproducible. Among the 61 examined papers, only three demonstrated effective solutions: CE-Net, TransResUNet, and Lung VAE.

Evaluation of these lung segmentation methods based on dice loss and IoU metrics revealed consistent superiority of CE-Net across experiments. Notably, TransResUNet exhibited limitations, struggling to accurately localize the lungs in certain instances. The direct comparison of generated masks further emphasized the robust performance of CE-Net over both TransResUNet and Lung VAE.

These findings highlight the challenges in achieving reliable and consistent results in deep learning for segmentation tasks, underscoring the significance of methodological choices in model development. The code is available at `https://github.com/Hryniewska/lung-segmentation-on-X-rays`.

Acknowledgment

The authors would like to thank Jakub Brojacz for his valuable impact on the research.

References

[1] Reamaroon, N., Sjoding, M. W., and Derksen, e. a., Hanneke. Robust segmentation of lung in chest x-ray: Applications in analysis of acute respiratory distress syndrome. *BMC Medical Imaging*, 20(1):116, 2020. doi: 10.1186/s12880-020-00514-y.

[2] Liu, W., Luo, J., and Yang, e. a., Yu. Automatic lung segmentation in chest x-ray images using improved u-net. *Scientific Reports*, 12:8649, 2022. doi: 10.1038/s41598-022-12743-y.

[3] Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *MICCAI*, pages 234–241. 2015. doi: 10.1007/978-3-319-24574-4_28.

[4] Hinterstoisser, S., Lepetit, V., Wohlhart, P., and Konolige, K. On pre-trained image features and synthetic images for deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018.

[5] Candemir, S., Jaeger, S., Palaniappan, K., Musco, J. P., Singh, R. K., Xue, Z., Karargyris, A., Antani, S., Thoma, G., and McDonald, C. J. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Transactions on Medical Imaging*, 33(2):577–590, 2014. doi: 10.1109/TMI.2013.2290491.

[6] Jaeger, S., Karargyris, A., Candemir, S., Folio, L., Siegelman, J., Callaghan, F., Xue, Z., Palaniappan, K., Singh, R. K., Antani, S., Thoma, G., Wang, Y.-X., Lu, P.-X., and McDonald, C. J. Automatic tuberculosis screening using chest radiographs. *IEEE Transactions on Medical Imaging*, 33(2):233–245, 2014. doi:10.1109/TMI.2013.2284099.

[7] Stirenko, S., Kochura, Y., Alienin, O., Rokovyi, O., Gordienko, Y., Gang, P., and Zeng, W. Chest X-Ray analysis of tuberculosis by deep learning

with segmentation and augmentation. In *IEEE ELNANO*. 2018. doi:10.1109/elnano.2018.8477564.

[8] Çallı, E., Sogancioglu, E., van Ginneken, B., van Leeuwen, K. G., and Murphy, K. Deep learning for chest X-ray analysis: A survey. *Medical Image Analysis*, 72:102125, 2021. doi:10.1016/j.media.2021.102125.

[9] Shamir, R. R., Duchin, Y., Kim, J., Sapiro, G., and Harel, N. Continuous dice coefficient: a method for evaluating probabilistic segmentations. 2019.

[10] Selvan, R., Dam, E. B., Detlefsen, N. S., Rischel, S., Sheng, K., Nielsen, M., and Pai, A. Lung segmentation from chest X-rays using variational data imputation. *Artemiss workshop at ICML*, 2020.

[11] Reza, S., Amin, O. B., and Hashem, M. TransResUNet: Improving U-Net Architecture for Robust Lungs Segmentation in Chest X-rays. In *TENSYMP*, pages 1592–1595. 2020. doi:10.1109/TENSYMP50017.2020.9230835.

[12] Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., Zhang, T., Gao, S., and Liu, J. CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Transactions on Medical Imaging*, 38(10):2281–2292, 2019. doi:10.1109/TMI.2019.2903562.

Exploring EEG-based age prediction: systematic biases, and disease-specific impacts

Martyna Poziomska[0000−0002−3754−7515],
Jarosław Żygierewicz[0000−0002−7536−0735],
Jacek Rogala[0000−0001−5298−920X]

[1] *University of Warsaw*
Faculty of Physics
Pasteura 5, 02-093 Warsaw, Poland
martyna.poziomska@gmail.com

Abstract. *Features of EEG signals change with age and health condition of a subject. In this contribution, we report the experiments concerning predicting the age of healthy and diseased subjects. The age regression model, trained on healthy individuals aged 1–90, exhibits systematic errors correlated with age. Analysis of Shapley values suggests the significance of coherence in higher beta and gamma bands for elucidating model explanation. Model predictions for patients with ICD-10 diagnoses reveal consistent discrepancies across different diseases, persisting post-correction of the systematic errors.*
Keywords: *EEG, artificial intelligence*

1. Introduction

Electroencephalogram (EEG) spectral properties and connectivity changes with age [1, 2]. It also reflects degenerative processes in the brain [3, 4, 5] and is sensitive to medication [6, 7].

Developing an explainable model that predicts a person's age based on the EEG signal can help to identify changes in specific signal features that cause abnormal deviations in age prediction. To achieve this, we propose using a gradient boost model that employs hand-crafted features extracted from the EEG signal. Detecting and ultimately understanding the cause of age prediction deviations that exceed normal levels may aid in the identification of new biomarkers for disease processes and the creation of personalized healthcare strategies.

DOI:10.17388/WUT.2024.0002.MiNI

2. Data

The utilized dataset, courtesy of Elmiko Ltd., comprises standard clinical EEG recordings from 8849 healthy individuals and 607 samples from patients with known medical conditions described by ICD-10 codes (Table 1). EEG signals were recorded using a standard 10-20 system of electrodes, referenced to a common average, and sampled at a sampling frequency of 100 Hz. The age range of subjects in the dataset is 1 to 90 years, with a uniform age distribution achieved by subsampling the original, larger dataset.

Table 1. Number of cases of different patient conditions according to ICD-10 codes

ICD-10 Code	**Count**	**Description**
G40	168	Epilepsy
I63	120	Cerebral infarction
G45	116	Transient cerebral ischemic attack
R55	81	Syncope and collapse
F43	70	Adjustment disorder
G98	50	Other disorders of the nervous system
R56	45	Other and unspecified convulsions
I64	41	Stroke, not specified as hemorrhage or infarction
normal	8849	Normal condition (reference category)

3. Methods

EEG signals were band-pass filtered in the 0.5–40 Hz frequency range. Continuous recordings were segmented into 6-second frames. Frames containing flat signals or voltages exceeding $600\mu V$ were rejected from further analysis. For each recording, we computed the following features:

- Signal power in frequency bands (f_b): {[0.5, 2], [1, 3], [2, 4], [3, 6], [4, 8], [6, 10], [8, 13], [10, 15], [13, 18], [15, 21], [18, 24], [21, 27], [24, 30], [27, 39], [30, 40]} Hz was computed for each channel and each frame, normalized such that the sum over channels and frequency bands equals one, and then aggregated across frames by median.

- Coherency between each pair of channels in the same frequency bands as above. Band-wise coherences between EEG channels were estimated as follows:

$$C_{xy}(f_b) = \frac{|\bar{S}_{xy}(f_b)|}{\sqrt{\bar{S}_{xx}(f_b) \cdot \bar{S}_{yy}(f_b)}}, \tag{1}$$

  where x, y – EEG channels, $S_{xx}(f_b)$ and $S_{xy}(f_b)$ are spectral and cross-spectral densities, respectively, estimated using the multitaper method [8] for one-second-long windows obtained from each frame, i.e., there were 6 estimates of densities per frame. $\bar{S}_{xx}(f_b)$ and $\bar{S}_{xy}(f_b)$ are the averages across the windows comprising given frame. Again, the coherency values obtained for each frame were aggregated by median to yield a coherency set for the given recording.

- Covariance matrices were computed for each channel, averaged across frames, and then transformed into tangent space. Tangent space projection of covariance matrix converts them into Euclidean vectors while conserving the inner structure of the manifold. After projection, vector-based classification can be applied.

The features described above, assembled into a vector, were the input to the regression model.

The training set consisted of data from 7329 healthy subjects, the test set of 1520 samples from healthy individuals, and 607 samples from patients with known medical conditions. The CatboostRegressor [9] model was designed and trained to predict each patient's age. This model type was selected due to its ability to handle complex relationships and produce accurate regression predictions in medical scenarios [10].

To determine the average impact of each feature on the model's predictions, we evaluated the mean value of absolute Shapley values across the test set. Since CatboostRegressor is constructed of trees, we employed TreeExplainer from the SHAP library [11].

4. Results

Figure 1 (bottom panel) illustrates residues, $\text{age}_{predicted} - \text{age}_{true}$, from the test set. Each dot corresponds to one recording. Colors correspond to condition clas-

sification according to Tab. 1. We observe that besides the random fluctuations of residua, there is a systematic deviation in prediction that changes with age.

Figure 1. Errors of age prediction as a function of age in the test set. Top panel: red – MAE for normal subjects, blue – MAE for normal subjects after accounting for the systematic error. Bottom panel: residues of the age prediction – each dot represents one individual from the test set. The black line is the median of the residues of normal subjects in 5-year bins, and the error bars – 95% confidence interval of the median.

The explanation of the model allows features to be ranked according to mean absolute Shapley values and, in this sense, estimates their importance in age regression. Figure 2 shows the average impact on model output magnitude for the 15 most important features. Most of them are coherences in gamma and beta bands.

To enable the use of the model in analyzing the discrepancies in age prediction in the diseased groups, we further analyzed the systematic error for normal recordings. Healthy patients from the test set were stratified into age subgroups every 5 years. For each subgroup, the prediction error was calculated as the difference between predicted and actual ages, and its median serves as a measure of systematic bias (the black line in the lower panel of Fig.1)

In the top panel of Figure 1, we observe that mean absolute error (MAE) changes with age non-monotonously. Subtracting the corresponding median from each age prediction stabilizes MAE (7.7±0.2 years) starting from age 15. This observation enables the investigation of regression behavior across various diseases

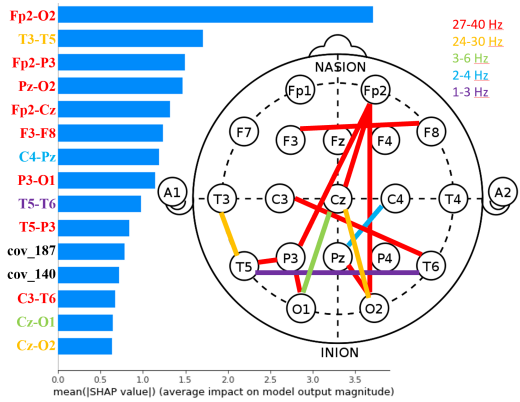


Figure 2. Explanation of the model. Bars indicate 15 features with the highest absolute value of Shapley values averaged across the test set. Most are coherences, and two are related to projections of covariance matrices into the tangent space. The font color of the named electrode pairs encodes the frequency bands according to the legend. The coherences displayed in the bar plot are also visualized on the topographic representation of the scalp with the same color code.

with the influence of systematic errors diminished. Figure 3 displays boxplots of age prediction errors in subjects grouped by ICD 10 codes. The notches in the boxplots indicate the 95% confidence intervals for the median. We observe significant variations across diseases, e.g., diseases coded as G40, R55, and R56 show an overestimation, whereas diseases coded as I63 and I64 result in age underestimation by the model. Considering the nature of these diseases, it's plausible that they correlate with physiological changes in the brain. These changes might manifest as coherence pattern alterations, leading to age estimation discrepancies.

5. Conclusions

The model exhibits a systematic error in age estimation, overestimating individuals below 50 and underestimating those above 50. Despite preliminary experiments with various models, including transformer analyzing raw EEG signals and SVM with covariances (not reported here), similar biases in the observed errors were noted. This suggests inherent biases in age prediction, likely related to the

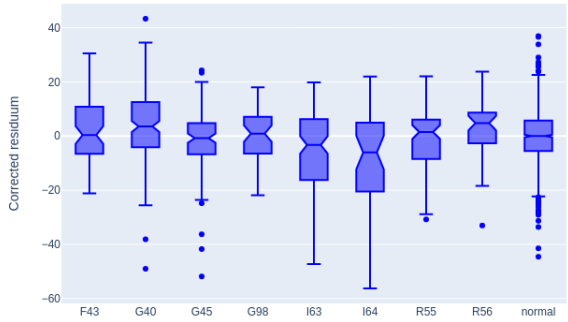


Figure 3. Notched box plot for bias-corrected regression error grouped by ICD 10 codes.

dataset or disease nature rather than model choice. However, these observations require further systematic cross-validation tests with other features to confirm such conclusions.

The utilized dataset has no information on potentially confounding factors like patients' lifestyles or genetic defects. These factors could also affect the EEG and observed bias, and should be considered in future studies.

Additionally, the SHAP explanation highlights the significance of EEG coherences, specifically in gamma and high beta frequencies, for age prediction. Bias-corrected regression errors reveal variations across diseases, with some leading to overestimation (e.g., G40, R55, R56) and others to underestimation (e.g., I63, I64). Diseases impacting cognitive function or neuronal activity may alter EEG coherence patterns, influencing age estimation. However, biases may be disease-specific or related to medications.

In conclusion, the systematic errors underscore the complexity of EEG-based regression models, which are influenced by physiological factors and dataset biases. Further research is needed to understand and address age estimation biases to refine the accuracy of EEG-based models in clinical applications.

References

[1] Roubicek, J. The electroencephalogram in the middle-aged and the elderly. *Journal of the American Geriatrics Society*, 25(4):145–152, 1977.

[2] Vysata, O., Kukal, J., Prochazka, A., Pazdera, L., Simko, J., and Valis, M. Age-related changes in EEG coherence. *Neurologia i Neurochirurgia Polska*, 48(1):35–38, 2014.

[3] Jeong, J. Eeg dynamics in patients with alzheimer's disease. *Clinical neurophysiology*, 115(7):1490–1505, 2004.

[4] Kowalski, J., Gawel, M., Pfeffer, A., and Barcikowska, M. The diagnostic value of EEG in Alzheimer disease: correlation with the severity of mental impairment. *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, 18(6):570—575, 2001.

[5] Shirahige, L., Berenguer-Rocha, M., Mendonca, S., Rocha, S., Rodrigues, M. C., and Monte-Silva, K. Quantitative electroencephalography characteristics for parkinson's disease: a systematic review. *Journal of Parkinson's Disease*, 10(2):455–470, 2020.

[6] Banoczi, W. R. How some drugs affect the electroencephalogram (EEG). *American Journal of Electroneurodiagnostic Technology*, 45(2):118–129, 2005.

[7] Höller, Y., Helmstaedter, C., and Lehnertz, K. Quantitative pharmacoelectroencephalography in antiepileptic drug research. *CNS drugs*, 32(9):839–848, 2018.

[8] Thomson, D. J. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982.

[9] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. 2019.

[10] Safaei, N., Safaei, B., Seyedekrami, S., Talafidaryani, M., Masoud, A., Wang, S., Li, Q., and Moqri, M. E-catboost: An efficient machine learning framework for predicting icu mortality using the eicu collaborative research database. *PLOS ONE*, 17(5):1–33, 2022.

[11] Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

Wavelet Packet Transform in Feature Extraction for EEG Signal Classification of Motor Imagery Tasks

Jan Rogowski[0000−0001−5467−7031], **Kacper Szmigiel**

Lodz University of Technology
Institute of Information Technology
al. Politechniki 8, 93-590 Łódź, Poland
jan.rogowski@p.lodz.pl

Abstract. *In this study the Wavelet Packet Transform is applied for feature extraction in EEG signal classification. The research was conducted on the EEG signal dataset for 52 subjects performing motor imagery tasks, such as imagining left/right-hand movement. The main goal of the work is to propose a feature vector that maximizes the imagining left or right hand movement classification quality. Performance of different classifiers is measured using accuracy scores for different wavelet transforms. The results of the classification tasks for combinations of wavelet transforms and classifiers are presented.*
Keywords: *EEG signal classification, Wavelet Packet Transform, motor imagery tasks*

1. Introduction

Electroencephalography (EEG) data is widely used in various fields for assessing brain function and activity. EEG is used in clinical settings to diagnose and monitor various neurological conditions, such as epilepsy, sleep disorders (e.g., sleep apnea, insomnia), brain tumors, and neurodegenerative diseases (e.g., Alzheimer's disease, Parkinson's disease). EEG serves as a fundamental component of the Brain-Computer Interfaces (BCIs), enabling direct communication between the brain and external devices. BCIs based on EEG allow individuals with severe motor disabilities to control assistive technologies, such as prosthetic limbs, computer cursors, and communication devices, using their brain activity.

DOI:10.17388/WUT.2024.0002.MiNI

EEG signals contain neural correlates of emotional states, allowing for the detection and classification of different emotions, such as happiness, sadness, fear, anger, and surprise. Machine learning algorithms are often employed to analyze EEG data and classify emotional states based on specific features extracted from the signals [1].

EEG in motor imagery tasks involves recording brain activity while individuals imagine performing specific motor movements, such as moving their hands or feet, without actually executing the movements. Users modulate their brain activity by imagining specific motor movements, and machine learning algorithms analyze EEG signals to classify the intended movement and generate corresponding control commands. Research in this area applies Common Spatial Patterns filtering [2, 3] and PARAFAC method [4] for feature extraction.

In this work, we were focused on applying the Wavelet Packet Transform in feature extraction for EEG data recorded during the execution of motor imagery tasks. Experiments are conducted to propose a feature vector that maximizes the imagining left or right hand movement classification quality.

2. Methods

2.1. Discrete Wavelet Transform and Wavelet Packet Transform

Discrete Wavelet Transform (DWT) is a powerful tool for analyzing signals in both time and frequency domains, providing valuable insights into the structure and characteristics of signals in various applications. It decomposes a signal into its constituent frequency components (wavelet coefficients), revealing both frequency and time information simultaneously. DWT provides a multiresolution analysis (MRA) of the signal, allowing for the representation of both coarse and fine details. By decomposing the signal at multiple scales, DWT captures information about different frequency bands and time intervals, making it useful for analyzing non-stationary and transient signals.

Wavelet Packet Transform (WPT) is an extension of the DWT. While DWT decomposes a signal into approximation and detail coefficients at different scales, WPT further decomposes the detail coefficients into subbands, resulting in a more detailed representation of the signal's frequency content. WPT decomposes a signal into a binary tree structure, where each node represents a subband at a specific frequency resolution. At each level of decomposition, the detail coefficients are split into subbands by applying additional wavelet transforms, resulting in a hier-

archical decomposition of the signal. WPT provides more flexibility and adaptability compared to DWT by allowing for a finer decomposition of the signal into subbands at different frequency resolutions. It offers a richer representation of the signal's frequency content, capturing both broad-scale and fine-scale features. Moreover it enables enhanced feature extraction from the signal by providing access to a wider range of frequency bands and details compared to DWT. Figure 1 shows the DFT and WPT signal decomposition trees.

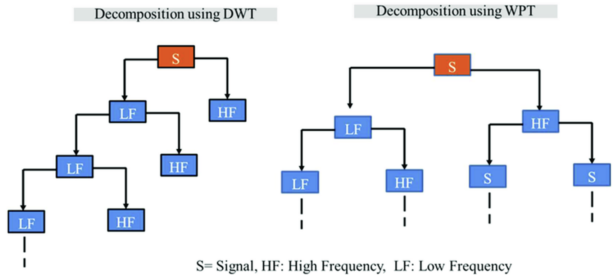


Figure 1. DWT and WPT signal decomposition

2.2. Motor imagery EEG dataset and 10-20 system

The 10-20 system is a standardized method used to describe the placement of electrodes on the scalp for EEG recordings. It provides a systematic way to position electrodes relative to specific anatomical landmarks on the head. Electrodes are positioned at specific percentages (either 10% or 20%) of the total distance between key anatomical landmarks on the head and labeled according to their position on the scalp using a combination of letters and numbers. The letters represent different regions of the head: F (frontal), C (central), P (parietal), T (temporal), and O (occipital). The numbers indicate the hemisphere and the specific location within each region. Odd numbers (1, 3, 5, 7, 9) typically denote positions on the left hemisphere, while even numbers (2, 4, 6, 8, 10) denote positions on the right hemisphere. Figure 2 shows the arrangement of electrodes in the 10-20 system.

Experiments are conducted on a dataset [5] that contains EEG signal recordings of 52 healthy subjects performing either motor or motor imagery hand movements. Each subject performed 100 or 120 trials for left/right-hand motor imagery movement. Each trial lasted around 7 to 8 seconds. 10 channel configuration in-

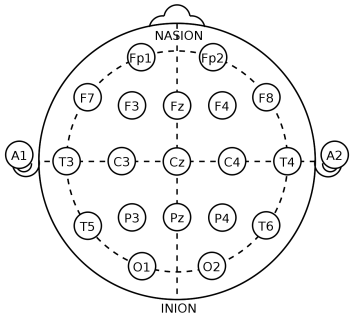


Figure 2. Electrodes arrangement in the 10-20 system

cluding FC3, C5, C3, C1, CP3, FC4, C2, C4, C6, and CP4 electrodes was used in the experiments.

EEG recordings are usually performed using a common reference placed somewhere on the head. The use of a reference electrode has been omitted by the method reference-free method called Common Average Reference (CAR) [2].

3. Experiments

For each subject, for each EEG channel, a 4-second signal segment was collected from each 7-8 second trial (epoch) corresponding to imagining left/right-hand movement (to get rid of, among other things, P300 distortion). For this section of the signal, WPT was calculated using the db2 and db4 wavelets (using other wavelets gave worse results). Then, only the WPT coefficients for the seventh level of decomposition were taken into account. These coefficients correspond to the frequency range 4-24Hz, which in turn corresponds to θ, α and β EEG frequency bands. In this way, 17 coefficients were obtained for each of the frequency subranges 4-6Hz, 6-8Hz ... 22-24Hz (10 such subranges in total). Then, for each such frequency subrange, 9 neighboring ones were selected from 17 coefficients and the sum of squares of these coefficients was calculated. Choosing the sum of squares as features allowed us to maximize the classification quality compared to using other calculated features. In this way, 10 features for each EEG channel were obtained for each subject for each epoch. A total of 100 features for 10 EEG channels. To obtain a larger number of elements in training and test

sets, the following procedure was applied. For each frequency subranges 4-6Hz, 6-8Hz ... 22-24Hz, for each iteration (9 iterations in total including the first one) 9 further neighboring coefficients were selected from 17 coefficients with a shift of one coefficient. And the feature vector was calculated as above. In this way, about 1800 feature vectors were obtained for each subject (200 epochs times 9 shifts for 9 coefficients out of 17). For some subjects, there are fewer than 200 epochs in the EEG dataset. Then, the feature vectors were fed to selected classifiers (kNN, SVM, LDA) using five-fold cross-validation. All calculations were performed in Python programming language.

4. Results

Figure 3 shows classification accuracy for kNN, SVM and LDA classifiers using the Daubechies 4 (db4) wavelet. As it can be seen, the best results are obtained for the kNN classifier for the two nearest neighbors. Figure 4 shows classification

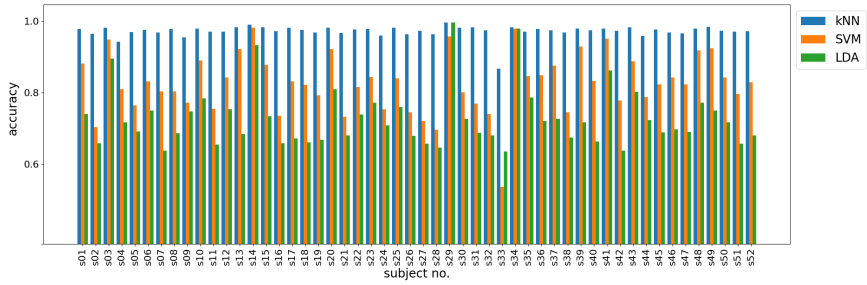


Figure 3. Classification accuracy for different classifiers and db4 wavelet

accuracy for kNN classifier for the two nearest neighbors using the Daubechies 2 (db2) and Daubechies 4 (db4) wavelets. The better results are obtained for the db4 wavelet.

The average classification accuracy for kNN classifier for the two nearest neighbors and db2 wavelet for all 52 subjects is 95% and for db4 wavelet is 97%.

Comparing the results presented in Figures 3 and 4 with the results obtained in [2] it can be concluded that a significant improvement in the quality of classification was achieved.

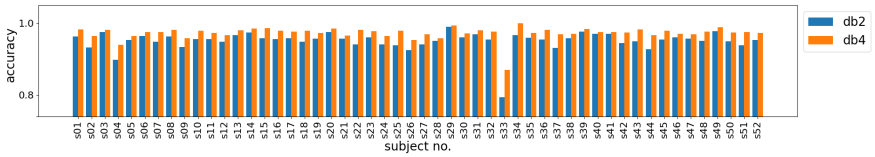


Figure 4. Classification accuracy, kNN classifier, db2 and db4 wavelets

5. Conclusions

The paper presents the application of WPT for feature extraction in EEG signal classification. Due to the use of the proposed solution, it was possible to improve the quality of the classification compared to the previously obtained results. The outcome of the classification tasks for different wavelet transforms and classifiers are presented.

References

[1] Dura, A., Wosiak, A., Stasiak, B., Wojciechowski, A., and Rogowski, J. Reversed correlation-based pairwised EEG channel selection in emotional state recognition. In *International Conference on Computational Science*, pages 528–541. Springer International Publishing, 2021.

[2] Zakrzewski, S., Stasiak, B., Klepaczka, T., and Wojciechowski, A. Vr-oriented EEG signal classification of motor imagery tasks. *Human Technology*, 18(1):29–44, 2022.

[3] Zakrzewski, S., Stasiak, B., and Wojciechowski, A. EEG-based left-hand/right-hand/rest motor imagery task classification. In *IEEE International Scientific Conference on Informatics*, pages 29–44. IEEE, 2022.

[4] Rosipal, R., Rostakova, Z., and Trejo, L. Tensor decomposition of human narrowband oscillatory brain activity in frequency, space and time. *Biological Psychology*, 169, 2022.

[5] Cho, H., Ahn, M., Ahn, S., Kwon, M., and Jun, S. C. EEG datasets for motor imagery brain-computer interface. *Giga Science*, 6:1–8, 2017.

Electrode Importance for EEG-based Schizophrenia Detection

Luís A. Alexandre[1][0000−0002−5133−5025],
Wlodzisław Duch[2][0000−0001−7882−4729]

[1]*Universidade da Beira Interior and NOVA LINCS*
Covilhã, Portugal
luis.alexandre@ubi.pt
[2]*Dept. of Informatics, Institute of Engineering and Technology, Faculty of Physics, Astronomy & Informatics, Nicolaus Copernicus University,*
Toruń, Poland
wduch@umk.pl

Abstract. *EEG is the preferred technique for objective diagnosis of mental disorders. Unfortunately, complex methods of EEG signal processing do not generalize well beyond a single data set and have not yet provided clinically useful biomarkers. Good biomarkers should be simple to interpret, use a minimal number of electrodes, and be based on short resting state EEG recordings. We have used the vector autoregressive EEG signal representation of two schizophrenia data sets, showing that a simple k-NN classifier provides state-of-the-art results with only 5 or 6 EEG channels.*
Keywords: *EEG, Resting State, Schizophrenia Diagnosis, Electrode Selection*

1. Introduction

EEG has been used as a simple means to detect several brain disorders and increase our understanding of the brain. In this paper, we focus on the relative importance of the electrodes used in EEG. We expect that results will strongly depend on the type of disorder. With a large number of electrodes and small data sets (typically less than 100 cases, including the control group), many input combinations may give similar accuracy. Although our method is general, We focus here

DOI:10.17388/WUT.2024.0002.MiNI

on the detection of schizophrenia using two popular data sets. The importance of identifying which electrodes have more discriminative power is twofold: first, it can help as a feature selection technique to use the best possible combination of EEG signal sources to improve the detection; second, it helps to identify which brain regions are involved in schizophrenia pathology.

A review of candidate biomarkers in psychiatric disorders has recently been published in [1]. Despite enormous investment in biomarkers based on neuroimaging, genetics, molecular and peripheral assays for autism, schizophrenia, anxiety disorders, major depression, bipolar disorder, substance use disorders, and PTSD, we do not have reliable biomarkers for objective diagnosis of patients. Many sophisticated methods have been proposed for diagnosis based on EEG, reviewed in [2], but they are too complex and thus difficult to use in clinical practice. Only simple and robust methods have a chance to be useful in practice and provide a real baseline for more refined approaches.

This paper is organized as follows: the next section presents a description of both the data used in this work and the methods used in the evaluation; the following section contains the experiments, followed by a discussion and some conclusions.

2. Methods and Data

To identify the most relevant electrode(s) for schizophrenia diagnosis, we will use a brute force approach that tries all possible electrode combinations and, for each of these, perform a k-fold cross-validation. The brute force method is done only once and ensures that we discover all sets of electrodes that give the best results. The data sets are not completely balanced; therefore, we will select the combinations that show the best F1 score for analysis.

2.1. Data Sets

For tests, we have selected the two most commonly used data sets [2]. The first [3], which we will refer to in the paper as data set A, contains measurements collected from 84 male adolescent subjects, 45 with schizophrenia, and the remaining 39 in the healthy control group. The data was collected using 16 electrodes at the 128Hz sampling rate.

The second data set [4], which we will call data set B, contains data from only 28 subjects, 14 control and 14 with schizophrenia. It was collected using 19

electrodes at the 250Hz sampling rate. To make the data sets similar in terms of the number of electrodes used and facilitate the comparison of results and the ability to obtain results in an acceptable time frame, we reduce the 19 electrodes in this data set to the same 16 used in data set A. We also subsample the data from 250Hz to 128Hz, and use only the first minute of data recordings for all subjects. This makes both data sets comparable in terms of the position of electrodes, sampling frequency, and signal duration. For both data sets, the only pre-processing done consisted of applying a high-pass filter to remove data below 0.1Hz.

2.2. Data Representation

The data is a set of N ($C \times S$) matrices, where N is the number of subjects, C the number of channels (electrodes), and S is the number of samples. Each of these matrices is represented using the Vector AutoRegressive (VAR) approach [5], by considering the data at each time step i as a random vector y_i, and modeling each subject's data at a given time step t as a combination of the previous L random vectors:

$$y_t = \nu + u_t + \sum_{i=1}^{L} A_i y_{t-i}$$

where the $C \times C$ matrices A_i and the $C \times 1$ vector ν contain the model's parameters. These will be used to represent the data. The value L is called the lag, and u_t is a zero mean random noise vector.

2.3. Classification

For each combination of electrodes, a k-fold cross-validation is used to evaluate the capability of the combination to distinguish between normal and schizophrenic patients. As a classifier, we used the 3-NN. It is very fast for small data sets, avoids draws for the two class problems, and provides deterministic results. Following the most used approaches in the literature, 5-fold cross-validation on data set A and 10-fold on the smaller data set B was used.

3. Experiments

Calculations were performed on a PC with an AMD Ryzen 7 3700X 8-Core processor, Pop!_OS 22.04, 32GB RAM, 1TB SSD and an NVidia RTX 3080TI

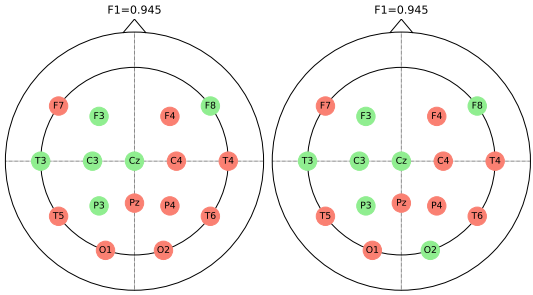


Figure 1. The two best subsets of electrodes for data set A, both with F1 score of 0.945. Note that they only differ in electrode O2.

GPU. The code was made in Python 3.10, and we used the libraries Scikit-learn for the classifier, Statsmodels for the VAR model, and Matplotlib for the figures. These calculations took around 40 hours to execute for each data set.

Searching for minimal sets of electrodes that reliably distinguish schizophrenic patients, we check all possible combinations of electrodes. For m electrodes, we conduct $2^m - 1$ k-fold cross-validation runs to obtain an estimate of the classification F1 score for each subset. For both data sets, we have 16 electrodes, so the total number of possible combinations is $2^{16} - 1 = 65535$ (all combinations with the exception of the case where no electrodes are used).

In data set A, out of the 65535 combinations, the best two yielded an excellent F1 score of 0.940. Figure 1 shows the electrodes used in these two subsets. For data set B, the three best subsets with F1 score of 0.897 were discovered. Classification using all electrodes (the original data set) gave an F1 score of 0.854 in data set A and 0.667 in data set B.

4. Discussion

Our goal was to establish a simple reference for more advanced machine learning approaches to the diagnosis of mental disorders. We have found that reducing 19 or 16 EEG channels to just five or six, and using the simplest classification method leads to excellent results. Comparison of results achieved with deep learning methods [2] shows that our simple approach is as good as any other method. Fusion of a vector autoregressive model, partial directed coherence, complex network measures of network topology to generate hand-crafted and learned features,

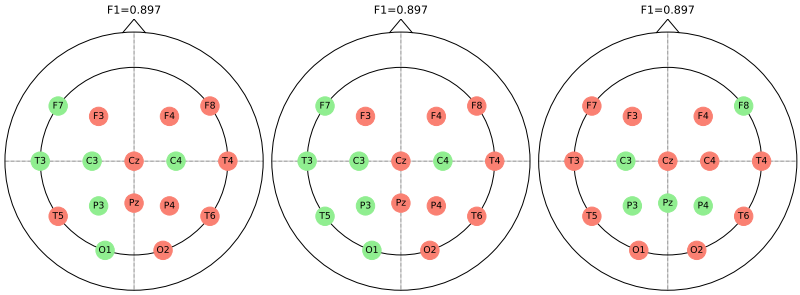


Figure 2. The three best subsets of electrodes for data set B, all with an F1 score of 0.897. Note that the two on the left and center only differ in electrode T5.

followed by three convolutional neural networks (CNNs) in 2730 dimensional input space, reaches 91.7±4.6% accuracy, or F1=0.93 [6]. Dozens of papers have reached significantly worse results than our simple benchmark calculations.

Positions of the electrodes, presented in Figures 1 and 2 for both data sets, may be justified based on the interpretation of differences in EEG power distribution between healthy controls and schizophrenic adolescents. We have already done a preliminary analysis of 5 classical frequency bands (delta-gamma), finding that in θ band F8, T3, T4, P4 electrodes are most important, in α bend F8, O2, Pz, and in β band P3 [7]. Unfortunately, we do not have space here to describe a detailed relationship of these findings with the vast literature on schizophrenia.

5. Conclusions

The main contribution of this paper is to show that a simple baseline approach should be performed before sophisticated deep-learning methods are applied to complex data. They frequently obfuscate the interpretation and are not more accurate than simple approaches. Therefore, it should be mandatory to compare results with simpler methods. Second, we have performed a selection of EEG channels showing that simple equipment, with just 5 electrodes, may provide accurate data. Here, we have analyzed two data sets commonly used to test methods for schizophrenia diagnosis. They are rather small, but in this field, it is impossible to find large data sets.

We will perform a similar analysis on other schizophrenia data sets, describe the relations of our findings to known brain processes that characterize schizophre-

nia, analyze power distributions in relevant brain regions, and use other EEG data sets to check if a simple reference model can also be used for other psychiatric conditions.

Acknowledgment

This work was supported by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP, Portugal, and the Polish National Science Center grant UMO-2016/20/W/NZ4/00354.

References

[1] Abi-Dargham, A., Moeller, S. J., Ali, F., and et. al. Candidate biomarkers in psychiatric disorders: state of the field. *World Psychiatry*, 22:236–262, 2023.

[2] Khare, S. K., Bajaj, V., and Acharya, U. R. Schizonet: a robust and accurate Margenau–Hill time-frequency distribution based deep neural network model for schizophrenia detection using EEG signals. *Physiological Measurement*, 44(3):035005, 2023.

[3] Borisov, S., Kaplan, A., Gorbachevskaya, N., and Kozlova, I. Analysis of EEG structural synchrony in adolescents with schizophrenic disorders. *Human Physiology*, 31(3), 2005.

[4] Olejarczyk, E. and Jernajczyk, W. EEG in schizophrenia, 2017. URL `https://doi.org/10.18150/repod.0107441`.

[5] Lütkepohl, H. *New Introduction to Multiple Time Series Analysis*. New York, Springer, 2005.

[6] Phang, C.-R., Noman, F., Hussain, H., and et. al. A multi-domain connectome convolutional neural network for identifying schizophrenia from EEG connectivity patterns. *IEEE J. of Biomedical and Health Informatics*, 24:1333, 2020.

[7] Duch, W., Tołpa, K., Ratajczak, E., Hajnowski, M., Furman, L., and Alexandre, L. A. Asymptotic spatiotemporal averaging of the power of EEG signals for schizophrenia diagnostics. Communications in Computer and Information Science, page 428–439. Springer Nature, Singapore, 2024.

Interpretable Methods for Sleep Staging: Insights from BErt-Inspired Neural Data Representation

Julia Jakubowska[1], **Jarosław Żygierewicz**[1]

[1] *University of Warsaw*
Faculty of Physics
Pasteura 5, 02-093 Warsaw, Poland
j.jakubowska10@student.uw.edu.pl

Abstract. *Polysomnography (PSG) is considered the golden standard for assessing sleep quality and diagnosing various sleep and consciousness disorders. Despite being a rapidly advancing problem in machine learning, automatic sleep staging based on PSG remains an open-ended discussion and frequently lacks comprehensive analysis. Using BErt-inspired Neural Data Representations, we explored the sleep staging classification process focusing on both the model's input and its latent space representations. The proposed model obtained state-of-the-art results. We established that known sleep EEG structures influence the model's decision-making process. The comprehensive results suggest that the model may be suitable for transfer learning and sleep staging.*
Keywords: *interpretable machine learning, polysomnography, sleep staging*

1. Introduction

Polysomnographic records (PSG) are a golden standard in sleep quality assessment and diagnosis of sleep and consciousness disorders. In clinical practice, recordings are still mainly accessed visually by experienced medical professionals, which is laborious, susceptible to subjectivity, and unreplicable even among the same medician.

Automating PSG data analysis and classification into wakefulness and sleep stages has been the subject of ongoing research, wherein machine learning has

DOI:10.17388/WUT.2024.0002.MiNI

also been integrated. Numerous approaches have been proposed [1]; however, it remains an open-ended discussion. Moreover, the frameworks mentioned above rarely undergo explanations, resulting in perceiving them as black boxes [2].

Our study aimed to improve the interpretability of models used for sleep staging tasks on PSG data. We evaluated the BErt-inspired Neural Data Representations (BENDR) [3]. The original work tested the model's suitability for transfer learning for specific tasks, including sleep staging, where authors achieved state-of-the-art results. However, despite this being a common practice in sleep research, they did not present separate classification accuracy for individual sleep stages, which is a fundamental starting point for further interpretation of the model's performance. Further, they did not pursue an interpretation of the model's decisions and its latent space, which could confirm the model's suitability for proper transfer learning. In this contribution, we explored the model's latent space using the t-SNE method and demonstrated the relevance of the Class Activation Map technique (CAM) [4] to analyze the model's input.

2. Data

We used an open-source subset of the Sleep-EDF database [5] from the PhysioNet databank [6], consisting of 36 PSG signals obtained from a study on temazepam effects on sleep. The subjects had mild difficulty falling asleep. Considering none of the subjects manifested any major sleep disorders and temazepam is not a high-impact medicine, we decided not to separate the dataset into placebo and temazepam subsets, which resulted in a more complex but larger training dataset.

All of the bioelectric signals were recorded with a sampling frequency of 100 Hz. Each PSG recording lasted about 9 hours and was performed during two subsequent nights, one after digesting a placebo and the other after temazepam intake. Each PSG recording included a corresponding hypnogram, which labeled each 30 s segment of the signal as one of the following stages: Wake, N1, N2, N3, and REM. The dataset provided only two EEG channels from the 10-10 system, Fpz-Cz, and Pz-Oz, and did not undergo any further preprocessing during the classification process.

3. Methods

The BENDR model [3] adapts a self-supervised speech recognition method wav2vec [7] to electroencephalographic data. It extracts EEG signal features in the time domain and creates their representation as a sequence of vectors. Following the wav2vec architecture, BENDR comprises two encoder submodels, one utilizing convolutional architecture while the other employs a transformer-based approach. Both parts of the model were first pre-trained on the non-target-specific, large-scale EEG dataset, which was supposed to make it suitable for transfer learning and fine-tuning to further specific tasks, including sleep staging.

In our contribution, we focused on exploring solely the convolutional part of the model with a linear classification layer, as such setup achieved the highest performance for most of the tasks presented by the authors. The submodel (conv-BENDR) is based on six convolution blocks, each composed of a 1D convolution layer, GroupNorm [8], and GELU activation. Each convolution consists of 512 filters and has a receptive field and stride length of 2, except for the first block, which has 3. The architecture effectively downsamples every 30 s fragment of the signal by a factor of 96, representing each fragment by 32 vectors of 512 features. We adjusted the model by changing its optimizer to RAdam [9], which resulted in a more regular behavior of the loss function for the current problem. Moreover, we adjusted the pre-classification layer for compatibility with other parameters essential for CAM analysis, which did not influence the model's performance. As a result, we obtained a 512-dimensional latent space representing each 30 s segment of the input signal.

We interpreted the model in the input space (PSG recording) and its latent space representation. The CAM technique [4] was used to explain the model in the input space. This technique relies on conserving the meaning of the temporal dimension of activation maps by the model architecture. It calculates a dot product between the model weights in the final layer of the encoder and those obtained after the Global Average Pooling [10] layer with softmax weights. The procedure generates vectors representing the significance of samples within the downsampled signal segments for each of the five classes considered in classification. The CAM representation is then upsampled to match the length of the input signal. This allows the significance of signal structures to be visualized and assessed based on the criteria established by the American Academy of Sleep Medicine (AASM) [11]. We used the t-SNE [12] to represent the high dimensional latent space of the encoder in a more comprehensive to the human eye 2D space.

4. Results

We fine-tuned the model on 32 PSG recordings and we evaluated the model on the dataset of 4, among which two recordings were collected after the placebo intake, and two after temazepam. The model's overall performance was consistent with the state-of-the-art results [2], obtaining BAC = 0.77, which exceeded the results presented by the authors [3]. Table 1 presents class-wise accuracy of classification across the test subjects for conv-BENDR. Achieved results are juxtaposed with the classification accuracy for a best-performing model reported in a previous study [13] utilizing the same dataset, but taking as input hand-crafted features. Both models demonstrated comparable performances, yielding similar outcomes in the classification of Wake (W) and REM classes. Nonetheless, there are observable differences in conv-BENDR effectiveness - the model notably excels in the classification of the N3 stage while exhibiting inferior performance in distinguishing N2 and N1 classes.

Table 1. Class-wise mean accuracy across test sets defined for conv-BENDR and best-performing model (XGB classifier) in [13].

Model	**Wake**	**N1**	**N2**	**N3**	**REM**
XGB classifier [13]	0.89	**0.6**	**0.92**	0.86	0.88
conv-BENDR	0.89	0.48	0.61	**0.99**	0.88

For a chosen test subject, we rendered a hypnogram predicted by the model and compared it with the expert's assessment (Fig. 1). The overall sleep architecture was maintained; however, it presented many irregularities and did not preserve the original smoothness in transitions between stages.

We applied the CAM method to identify critical input fragments for classification and assessed whether they align with structures and graphoelements established by AASM. By applying the CAM-based colormap, we identified segments where the model focused on structures akin to human scorers. In Figure 2.a), CAM revealed the recognition of N2-specific graphoelements, sleep spindles, which oscillate around 15 Hz. Rapid eye movement is a crucial criterion for determining REM sleep; in Figure 2.b) structures reminding of such movements are considered high-impact for the classification process. In Figure 2.c), N3 delta waves (slow oscillations) dominated model's decision, which is consistent with AASM criteria.

We performed a t-SNE analysis to visualize the reduced latent space of the

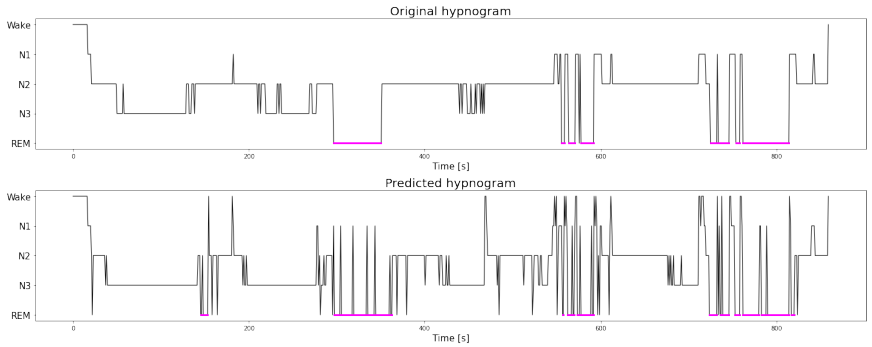


Figure 1. Sample hypnograms for one of the test subjects.

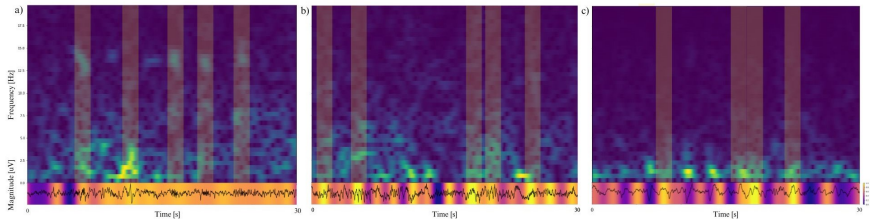


Figure 2. CAMs visualized for exemplary 30 s signal segments from Pz-Oz channel from PSG representing: a) N2, b) REM, and c) N3 sleep stages.

encoder part of the model. In Figure 3, we observe the expected clustering of stages aligning with well-established sleep physiology.

5. Conclusions

Obtained results reveal the state-of-the-art performance of the conv-BENDR model. Compared to the model [13] utilizing the same dataset [5], the presented architecture excels at the classification of the N3 stage.

The overall results demonstrate that explanations at the input space level allow us to validate the model against AASM recommendations. On the other hand, visualizations of the latent space reveal a reasonable behavior of the model and classification process as it mimics the overall sleep architecture and sleep stage

transitions. However, the results may benefit from applying a method utilizing information from adjacent signal segments to contextualize the predicted hypnogram, imitating the evaluation process of experienced human scorers.

Moreover, successful fine-tuning of the model to PSG data and achieving high classification and interpretation results indicate that the BENDR model may be suitable for transfer learning to perform sleep staging tasks despite the relatively limited-scale dataset.

Acknowledgment

The authors would like to thank Google Deep Mind for supporting the project as a part of the scholarship for graduate students of the University of Warsaw.

References

[1] Chriskos, P., Frantzidis, C. A., Nday, C. M., Gkivogkli, P. T., Bamidis, P. D., and Kourtidou-Papadeli, C. A review on current trends in automatic sleep staging through bio-signal recordings and future challenges. *Sleep medicine reviews*, 55:101377, 2021.

[2] Phan, H., Mikkelsen, K. B., Chén, O. Y., Koch, P., Mertins, A., and Vos, M. D. Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification. *CoRR*, abs/2105.11043, 2021.

[3] Kostas, D., Aroca-Ouellette, S., and Rudzicz, F. Bendr: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in human neuroscience*, 15:653659, 2021.

[4] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929. IEEE Computer Society, 2016. ISSN 1063-6919.

[5] Kemp, B., Zwinderman, A., Tuk, B., Kamphuisen, H., and Oberye, J. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

[6] Goldberger, A., Amaral, L., Glass, L., Havlin, S., Hausdorg, J., Ivanov, P., Mark, R., Mietus, J., Moody, G., Peng, C.-K., Stanley, H., and Physiobank, P. Components of a new research resource for complex physiologic signals. *PhysioNet*, 101, 2000.

[7] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc., 2020.

[8] Wu, Yuxin, and He., K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 2018.

[9] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond, 2021.

[10] He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, 2016.

[11] B., B. R., Brooks, R., Gamaldo, C., Harding, S. M., Lloyd, R. M., Quan, S. F., Troester, M. T., and Vaughn, B. V. Aasm scoring manual updates for 2017 (version 2.4). *Journal of clinical sleep medicine*, 13:665–666, 2017.

[12] van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[13] Lal, U., Mathavu Vasanthsena, S., and Hoblidar, A. Temporal feature extraction and machine learning for classification of sleep stages using telemetry polysomnography. *Brain Sciences*, 13(8), 2023.

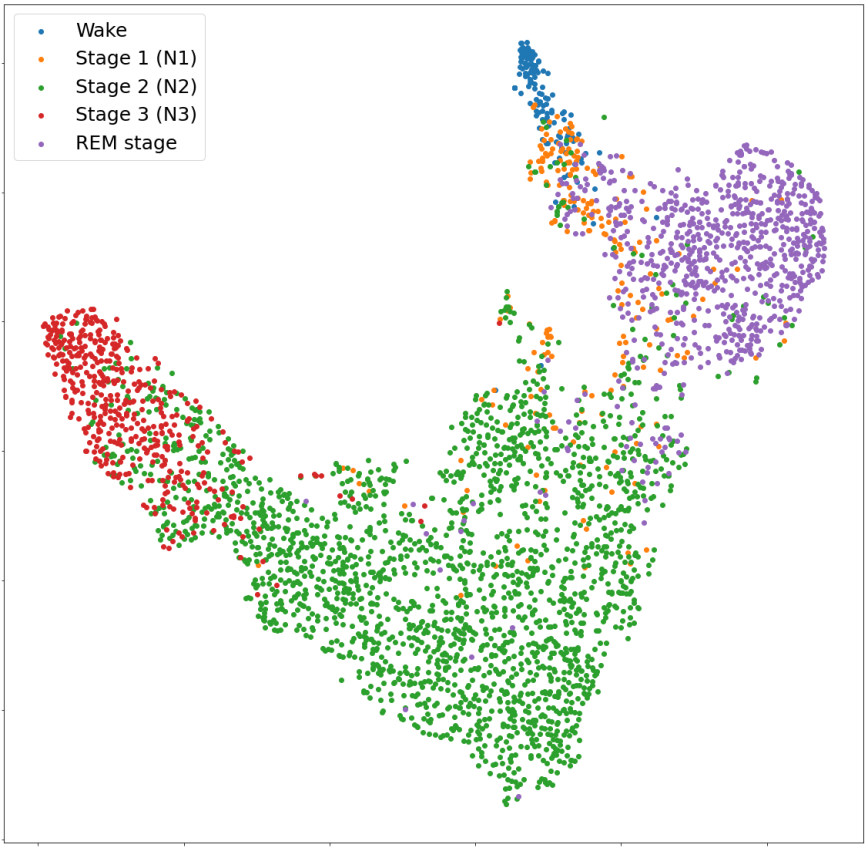


Figure 3. T-SNE visualization of embeddings of 30 s sleep EEG segments of the test dataset. Dot colors correspond to sleep stages from the expert's scoring.

Wavelet feature extraction: the case of dermatoscopy imaging

Grzegorz Surówka[1][0000−0001−5110−4238]

[1]*Jagiellonian University*
Faculty of Physics, Astronomy and Applied Computer Science
Łojasiewicza 11, 30-348 Kraków Poland
grzegorz.surowka@uj.edu.pl

Abstract. *We present an example of a wavelet base, RBio-3.1, which can be successfully used for feature extraction for the classification of dermatoscopy images. Wavelet feature extraction provides features based on the pixel energy content of the Mallat WPT decomposition filters. Such features can serve as high-performance, resolution-robust wavelet features for melanoma. Feature extraction and feature selection are important factors in the data analysis pipeline.*
Keywords: *machine learning, feature extraction, feature selection, wavelets, computer-aided diagnosis*

1. Introduction

Rapid prototyping in machine learning offers several benefits. It allows for experimentation with different models, their hyperparameters, and/or learning features. This is the first step before developing production/optimal models. Prototyping reduces the time of development, gives a deeper understanding of the problem domain, and allows for exploring new ideas, which may be potentially advantageous for the problem.

Feature engineering is a crucial step in the process of preparing data for machine learning. Good features in supervised learning should be highly relevant to the target variable and not relevant for background/noise, should be stable, and give insights into the phenomena that produced the data.

DOI:10.17388/WUT.2024.0002.MiNI

In medical imaging, feature engineering is essential to extract meaningful information, enabling the identification of structures, lesions, or abnormalities. Medical doctors examine human skin with the naked eye, but more often, with a dermatoscope, which is the gold standard for dermatological diagnosis.

This study presents research on the extraction of wavelet features from dermatoscopy images of pigmented skin lesions for computer-aided skin cancer diagnosis (CAD). This is an alternative way to analyze skin texture from dermatoscopy images. The segmentation-like approaches assume that skin patologies, e.g. melanoma, the most lethal human cancer, can be visually examined based on shape, color, and structure content. Unfortunately, if there are no classic tumor structures, if the image has noise or artifacts, poor illumination or resolution, these methods usually fail [1].

The melanoma CAD systems have been developed for years (e.g. [2]). They help in the early diagnosis of skin lesions, reduce the time and cost of treatment, and support non-invasive methods in dermatology.

Deep learning methods, specifically convolutional neural networks (CNNs), have gained significant attention and achieved remarkable success in various image-related tasks. In addition to their model efficiency, CNNs can automatically learn features from the raw pixel values of an image [3]. However, shallow (machine) learning techniques remain useful and applicable in scenarios where deep learning may not be the optimal choice. There are four such cases for image classification tasks:

- When the dataset is small: Shallow learning methods can generalize better to new, unseen data when the dataset size is limited.

- When computational resources are limited: Shallow learning techniques are suitable for resource-constrained environments where computational power is restricted.

- When feature engineering is crucial: In fields like medicine, where understanding the relationships between input features and output predictions is critical, shallow learning methods can be advantageous.

- When specialized features are required: Shallow learning can be beneficial for extracting specific features, such as resolution-robust features. The early detection of cutaneous melanoma and other pigmented skin lesions often involves small datasets and limited computational resources, particularly in

scenarios like self-diagnosis on mobile devices. Additionally, the lack of access to high-resolution images is another reason to construct features that can be informative at lower resolutions.

Overview articles on the state-of-the-art of melanoma skin lesion classification by different machine learning paradigms can be found, e.g. in [2, 4, 5].

2. Wavelet feature extraction

Our study is based on the MelanomaML dataset [6], which consists of high-resolution JPEG dermatoscopy images with a resolution of 2272×1704. To investigate the robustness of wavelet features towards resolution reduction, we additionally created a low-resolution dataset of 568×426 (by averaging pixels in 2×2 blocks twice). One more iteration of image degradation to a resolution of 284×213 was tested, but it turned out that at such a resolution, the learning metrics collapse (e.g. AUC is around a maximum of 60%). It appears that at this resolution, the skin texture analyzed by the wavelets becomes too coarse.

Preprocessing of the images included noise reduction (filtering) and removal of artifacts (such as hair and liquid bubbles). As a next step, lesion and background segmentation was performed [7].

The wavelet transform was applied to the images as filters [8], decomposing them into four sub-images: one coarse sub-image (LL), one sub-image of details in the rows (LH), one sub-image of details in the columns (HL), and a sub-image of details in both directions (HH). In Wavelet Packet Transform (WPT) [9], this operation is performed recursively on all channels, and products of the transform are downsampled by two at each iteration. We produced two iterations of WPT, yielding $1 + 4 = 5$ different transformation branches.

We assumed a rotation-invariant form of the feature set used in each transformation branch. Our aim was to make the analysis independent of two factors: the orientation of the mole in the dermatoscopy image and the orientation of the mole in relation to the skin texture (e.g., pigment network). To meet these requirements, we extracted the following simple, energy-based features [10]:
energies of the sub-images: e_i, $i = 1, 2, 3, 4$ (4 features),
maximum energy ratios: e_i/e_{max} (4 features), and
fractional energy ratios: $e_i/\sum e_k$ (4 features).
Energy e_i was defined as the sum of absolute values of the pixels.

When 5 WPT branches are applied, altogether 60 features are produced. Our goal was to compare the efficiency of features extracted from different wavelet families. These families were taken into account: Daubechies, Symlets, Coiflets, Biorthogonal, and Reverse Biorthogonal wavelets [11]. Several earlier experiments, e.g., [12], have shown how different wavelets contribute to the classification efficiency of dermatoscopy images and how they behave in reduced image resolutions.

Drawing conclusions from these experiments, in this article we present a wavelet from the Daubechies family (Daubechies-6), which is usually chosen for machine learning experiments of pigmented moles for its well-established construction and extensive literature, and a wavelet from the Reverse Biorthogonal (RBio) family (RBio3.1), which demonstrates excellent properties in maintaining high classification performance as image resolution decreases.

Since our feature extraction approach is based on machine learning models (wrappers), we used nine different model prototypes of different paradigms of learning. They are:
(1) Decision tree (Gini),
(2) Naive Bayes with the Gaussian kernel,
(3) KNN with the cosine distance metric optimized for the number of neighbors,
(4) Multi-Layer Perceptron with two hidden layers and regularization strength (lambda) of 0.01,
(5) Discriminant analysis with a linear kernel,
(6) Logistic Regression,
(7) Support Vector Machine (C-SVM) with a Gaussian kernel,
(8) Bagged trees with 30 weak learners,
(9) Adaboost with 30 weak learners (learning rate=0.1).

The models were roughly optimized using Bayesian optimization (30 iterations), focusing only on the most important hyperparameters. These include: the kernel (SVM, NBC), learning rate (AdaBoost), number of neighbors (kNN), number of splits (decision trees, AdaBoost), distance metric (kNN). In ensemble models, 30 component models were used. For MLP, the network was limited to two hidden layers.
Our goal was not to develop well-trained production classifiers but to analyze how the wavelet features are suited to the classification of dermatoscopy images under different image resolutions.

Our models were trained with 30 features selected from the full set of 60 features with the help of the Minimum Redundancy Maximum Relevance algorithm

(mRMR) [13], a very popular and multivariate filter-based algorithm with mutual information as the metric. mRMR addresses the problem of relevance of a given attribute and its redundancy (correlation/dependency) simultaneously.

In clinical practice, Positive Predictive Value (PPV) and Negative Predictive Value (NPV) are considered particularly important metrics, because they provide valuable insights into the accuracy and reliability of diagnostic tests/procedures. A high PPV helps clinicians avoid unnecessary interventions in cases where the positive test result is less reliable. A high NPV, on the other hand, assures clinicians that a negative test result is highly reliable in ruling out the presence of the disease. Although PPV and NPV are considered the best from a clinical classification standpoint, they are not utilized in the CAD (Computer-Aided Diagnosis) of melanoma. AUC (area under the ROC curve) and, less frequently, accuracy are typically utilized when reporting progress in this field. In this study, we use a single metric, AUC.

The learning metric was calculated in 10-fold cross-validation.

The results can be found in Fig. 1. As we can see:

- Wavelet RBio 3.1 [11] exhibits greater variability across models, but in six out of nine models, it achieves better results (by approximately 10%) than the popular (sometimes "standard" or "first choice") Daubechies wavelet. This difference, though seemingly small, is significant. The choice of a wavelet affects the height of the training metric if the goal is to train a model.

- The performance of models on images with four times reduced resolution (16 times fewer pixels!) should, in principle, be lower (red line). We observe this for the Daubechies wavelet, while the RBio3.1 wavelet shows even greater classification ability for this low resolution. This is of great importance, not only for the efficiency of the machines on which such classification is made (e.g., ARM-based mobile devices) but also for data acquisition. It turns out that for certain wavelet features, the models have performance comparable to that for large images. The visible gap between performance in both resolutions may be larger for unfavorable cases of wavelets that do not analyze the skin texture well. Daubechies wavelets are quite flexible in this respect. Importantly, good performance of the feature subset for different resolutions, considering the image analysis, is achieved for a small number of features (30).

The result presented here for the RBio3.1 wavelet family confirms its excellent performance for the classification of pigmented skin lesions (melanoma, dysplastic nevi), as discussed in [[12, 14, 15]]. This wavelet belongs to the family of Reverse Biorthogonal wavelets, among which several others also demonstrate high classification efficiency. Similarly, biorthogonal wavelets exhibit high classification accuracy. Exploring a wide range of wavelet families allows us to select a wavelet that initially boosts classification effectiveness by 5-10%. The purpose of the study presented here was to demonstrate such effectiveness without a significant impact from model optimization.

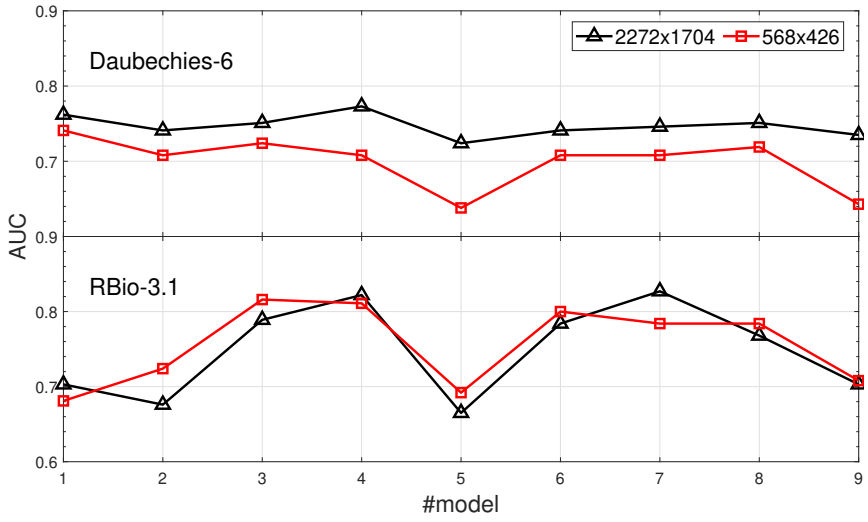


Figure 1. Efficiency (AUC) of simple model prototypes on wavelet features derived with WPT using the base wavelets of Daubechies-6 and Reverse Biorthogonal 3.1 wavelets. Model prototypes are introduced in Section 2. Points have been connected with a line to highlight the level and trend of the metric (AUC).

3. Conclusions

Efficiency aspects of any machine learning experiment, such as regularization, the minimum data requirements, etc., are important factors that should be carefully planned. In CAD, practical aspects play a role, so performance is the ultimate goal (considering time and memory requirements). The ability to select lower resolution images without significantly reducing classification performance is invaluable.

Our research not only contributes to melanoma feature extraction but also addresses a more general feature extraction and feature selection task. This task aims to derive efficient and robust features for the analysis of demanding textures, such as those found in digital imaging.

References

[1] Maglogiannis, I., Zafiropoulos, E., and Kyranoudis, C. Intelligent segmentation and classification of pigmented skin lesions in dermatological images. *Lect. Notes Comp. Sci.*, 3955:214–223, 2006.

[2] Dick, V., Sinz, C., Mittlbock, M., Kittler, H., and Tschandl, P. Accuracy of computer-aided diagnosis of melanoma: A meta-analysis. *JAMA Derm.*, 155(11), 2019.

[3] Surówka, G. Transfer learning from imagenet to the domain of pigmented nevi. In L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. Zurada, editors, *AISC*, pages 263–273. Springer Nature Switzerland, 2023. ISBN 978-3-031-42505-9.

[4] Kumar, Y., Gupta, S., Singla, R., and Hu, Y.-C. A systematic review of artificial intelligence techniques in cancer prediction and diagnosis. *Arch. Comp. Meth. Eng.*, 29(4):2043–2070, 2022.

[5] Hasan, K., Ahamad, A., Yap, C., and Yang, G. A survey, review, and future trends of skin lesion segmentation and classification. *Comp. Biol. Med.*, page 106624, 2023.

[6] ML, M. doi.org/10.17026/dans-zue-zz2y, Last accessed 01 December 2021. URL `https://doi.org/10.17026/dans-zue-zz2y`.

[7] Surówka, G. and Ogorzałek, M. Segmentation of the melanoma lesion and its border. *Int. J. Appl. Math. Comp. Sci.*, 32(4):683–699, 2022.

[8] Mallat, S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pat. Anal. Mach. Intel.*, 11(7):674–693, 1989.

[9] Posch, T. The wave packet transform (wpt) as applied to signal processing. In *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pages 143–146. IEEE, 1992.

[10] Patwardhan, S., Dhawan, A., and Relue, P. Classification of melanoma using tree structured wavelet transforms. *Comp. Meth. Progr. Biomed.*, 72:223–239, 2003.

[11] Kovacevic, J. and Sweldens, W. Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Image Proc.*, 9(3):480–496, 2000.

[12] Surówka, G. and Ogorzałek, M. Resolution invariant wavelet features of melanoma studied by SVM classifiers. *Plos One*, 14(2):e0211318, 2019. doi:https://doi.org/10.1371/journal.pone.0211318.

[13] Bugata, P. and Drotar, P. On some aspects of minimum redundancy maximum relevance feature selection. *Sci. China Inf. Sci.*, 63:1–15, 2020.

[14] Surówka, G. and Ogorzałek, M. Naive bayes learning of dermoscopy images. In *Art. Int. Soft. Comp.*, pages 294–304. Springer, 2019.

[15] Surówka, G. and Ogorzalek, M. Wavelet-based logistic discriminator of dermoscopy images. *Exp. Syst. Appl.*, 167:113760, 2021.

Generative Adversarial Network Training for Image Reconstruction in Capacitively Coupled Electrical Impedance Tomography

Mikhail Ivanenko[0009−0006−8682−2751], **Damian Wanta**[0000−0002−1596−6524], **Waldemar T. Smolik**[0000−0002−1524−5049], **Przemysław Wróblewski**[0000−0002−6713−9088], **Mateusz Midura**[0000−0002−2449−0652]

Warsaw University of Technology
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
Damian.Wanta@pw.edu.pl

Abstract. *Electrical impedance tomography (EIT) aims to visualize a distribution of electrical properties inside an object based on non-invasive impedance measurements. Capacitively coupled impedance electrical tomography (CCEIT), a new contactless technique based on capacitance measurements, can improve patient comfort and measurement quality. The promising image reconstruction approach in EIT is an application of artificial neural networks (ANN). However, there are three essential questions about supervised ANN training: network architecture, training procedure, and size of the training dataset. In this work, we focus on a Pix2Pix-inspired approach for training a conditional generative adversarial network (cGAN) with a U-Net-based generator. The evaluation of the network was conducted using a numerical simulation of tomographic measurements of human head and thorax numerical phantoms. The sensors with 16 and 32 large-area electrodes were simulated using the ECTSim toolbox for Matlab. Reconstruction quality assessment was conducted by calculating pixel-to-pixel metrics as a function of the validation dataset.*

Keywords: *Electrical impedance tomography, neural networks, image reconstruction, inverse problem, cGAN, supervised machine learning, deep learning, numerical simulation, training set size, Pix2Pix*

DOI:10.17388/WUT.2024.0002.MiNI

1. Introduction

Electrical impedance tomography (EIT) is a non-invasive and low-cost tomographic technique that reconstructs electrical properties inside an examined object based on impedance measurement between electrodes surrounding that object. In contact-less capacitively coupled EIT (CCEIT), large-area electrodes are used to measure complex-valued capacitance to avoid a variable electrode-skin resistance, negatively influencing measurement accuracy [1], [2]. CCEIT can be used in medical diagnostics for the initial, early detection of pathological changes in a lung or brain due to the significant variation of tissues' electrical properties [3]. Conventional EIT devices have already been introduced into clinical practice for lung ventilation and perfusion monitoring in intensive care units [4], [5]. There is constant research in the direction of proving EIT feasibility for other possible applications, for example, for brain imaging [2], [6].

The image reconstruction problem in CCEIT is an ill-posed, severely numerically ill-conditioned nonlinear inverse problem [7]. There are many methods of inverse problem solving in EIT, including methods based on an approximation of analytic solution (Calderon or dBar method) [8], [9], linear algebraic methods, like one-step LBP or iterative Landveber algorithm [10], [11] [12], or nonlinear optimization methods, like iterative Levenberg-Marquardt [13], [14]. These methods only allow us to obtain an approximation of the solution, and in the case of nonlinear methods, they require a long computation time. Determining the tomographic model in the form of a Jacobi matrix requires solving the electric field equation in the tomographic sensor, which must be performed in each step of the nonlinear algorithm. An alternative to the classical approach, machine learning methods require less computational power at the moment of reconstruction. Access to increased computing power in recent years has led to great interest in tomographic image reconstruction based on machine learning [15]–[18]. A wide variety of ML-based algorithms proposed in EIT image reconstruction includes Artificial Neural Networks, Random Forests, K-Nearest Neighbors, Elastic Net, Ada Boost, and Gradient Boosting [19]. Our previous work has shown a conditional generative adversarial ANN (cGAN) advantage over fully-connected ANN in CCEIT image reconstruction [20].

The main challenge in ANN supervised learning is acquisition of a sufficiently large training dataset. Its creation using numerical simulation is a very time consuming. Therefore, arises the question on how large should be the training dataset to successfully train the neural network. The main goal of the study is to explore

how training dataset size influences the reconstruction quality.

In this paper, we discuss the cGAN training procedure and choose the training dataset size for supervised learning in the context of solving the CCEIT inverse problem. Due to the nature of the CCEIT model, the number of output nodes in the trained network is much greater than the number of input nodes.

2. Materials and methods

2.1. Neural network architecture

In this work, we focus on cGAN architecture (Figure 1) [21], consisting of a generator and discriminator, where the generator is intended to generate an image based on the provided condition, and the task of the discriminator is to detect if the image appears to be generated or taken from a training set.

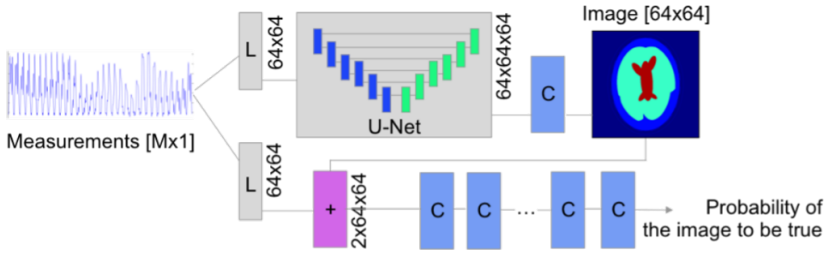
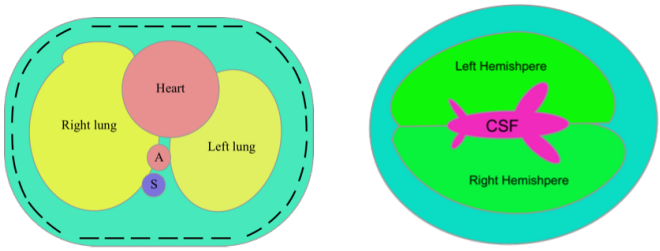


Figure 1: Network architecture: L – linear layers, C – convolution layers.

The generator for EIT image reconstruction was based on U-Net architecture [22]. In our work, we used convolutional blocks consisting of a convolutional layer with a kernel of size 3, padding 1, and stride of 2 connected with an optional batch normalization [23] layer and leaky rectified linear unit (ReLU) activation function. In deconvolutional blocks, we used a transposed convolutional layer with kernel of size 2, and stride 2. It is connected with a batch normalization layer and an optional dropout of 50% of connections. As an activation function, we use the same ReLU activation. The resulting generator architecture for 64x64 image reconstruction consists of six convolution and six deconvolution blocks followed by a convolutional layer intended to reduce U-Net output to a 2D image. A fully connected layer at the beginning of the network was added to pass the measurement



(a) (b)

Figure 2: Numerical phantoms: a) human thorax: A – aorta, S – spine b) human head with two brain hemispheres and cerebrospinal fluid (CSF).

data vector to the generator input. As a discriminator, we used a classifier consisting of three convolutional blocks of the same construction as used in the generator. The convolutional layer (kernel size 4, padding 1, stride 2), followed by the batch normalization layer and ReLU activation function, was attached to the last convolutional block. An additional convolutional layer (kernel size 4, padding 1, stride 3) was attached with a sigmoid activation function, which ensures the output was in the range [0;1].

2.2. Dataset generation

In the context of the CCEIT inverse problem, the training dataset should contain samples of electrical properties distribution in the object and corresponding expected capacitance measurements. Collecting thousands of medical examinations is a very complicated task, but it is possible to conduct a numerical simulation. We used the ECTSim toolbox for MatLab [24] to generate training datasets using developed numerical phantoms of the human head and thorax. Phantoms were made to represent tissue distribution using a combination of ellipses, as shown in Figure 2. We also introduced two medical conditions significantly affecting tissue conductivity to appear in the datasets: pneumothorax and pleural effusion for the thorax dataset and ischemic and hemorrhagic stroke for the head dataset. Values of electrical permittivity and conductivity were assigned for a given tissue according to [3]. For the thorax examinations, 150,000 samples representing healthy and illness-affected lungs were generated, and 44,000 samples representing random ellipses. For the head examinations, 200,000 samples representing

healthy and illness-affected brains were generated. For validation purposes, 42,500 and 50,000 additional samples were generated.

2.3. Neural network training procedure

Classical cGAN training supposes using a random latent vector to provide necessary intra-class variability and simple binary cross entropy (BCE) loss function. This approach works well with relatively small output image sizes. However, increasing the output image size to 64x64 causes the network to lose stability. Inspired by the Pix2Pix approach intended to train cGAN on large images [25], we modified the training procedure in the following way: we removed the latent vector in favor of dropout layers in convolutional blocks; we modified the generator loss function to include L1 and L2 loss components alongside with a classic BCE loss; we used different learning rates changing according to cosine annealing schedule. Proposed approach allows to easily balance generator and discriminator strengths, preventing the discriminator from collapsing. In the normal working mode, the discriminator loss is supposed to tend to one-half, meaning that the discriminator cannot distinguish generated images from those presented in the training set. If the discriminator is going to collapse, then the loss starts going to zero. In such a case, it is possible to adjust the discriminator's learning rate to weaken the discriminator's classification ability. To increase network robustness, we introduced a Gaussian noise into measurement data with SNR equal to 30 dB. In the case of the 32-electrode thorax model, we used only an imaginary capacitance component to train the network, having the network's input be a vector of size 992. In the case of the 16-electrode head model, we used both complex capacitance components as separate real numbers, with the input being a vector of size 480. As target images, we used conductivity distribution. The validation procedure includes comparing reconstructed images with given electrical properties distribution and calculating the metric's mean value over the validation dataset, not including samples in the training dataset. We used simple pixel-to-pixel metrics for comparison, such as structural similarity index (SSIM) and 2D correlation coefficient (CC).

3. Results

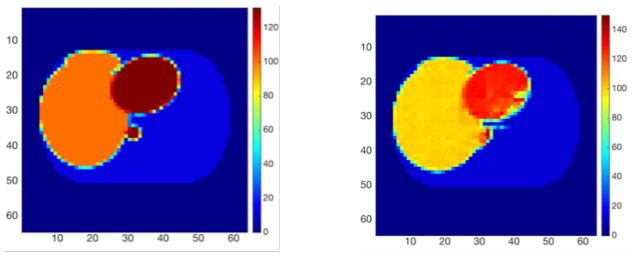
Using the described approach, we successfully trained two cGAN models aimed at reconstructing thorax and brain CCEIT images. After training we validated model generalization ability by reconstructing the sample representing the

edge case not presented in the training dataset (Figure 3). We conducted the validation by adding Gaussian noise to the measurement data, providing a signal-to-noise ratio (SNR) of 30 dB and 10 dB. Calculated mean metrics values are shown in Table 1.

For the human brain model, we conducted additional experiments and trained the network on a randomly selected training dataset subset and evaluated reconstruction quality using the noisy and very noisy measurement data. The value of the image metric in the function of the training dataset size is shown in Figure 4.

Table 1: The mean value of image quality norm in the validation dataset

SNR, dB	Thorax model		Brain model	
	SSIM	CC	SSIM	CC
30	0.87	0.98	0.81	0.95
10	0.82	0.95	0.71	0.92



(a) (b)

Figure 3: Conductivity reconstruction of thoracal phantom representing the patient with only one lung: a) ground truth b) reconstructed image.

4. Discussion and further development

Two trained different cGAN models were able to reconstruct CCEIT images. The very noisy measurement data doesn't substantially worsen image metrics. The networks of this architecture, trained using the Pix2Pix approach, might be feasible to reconstruct images from real measurement data. While analyzing the dependence of the reconstruction quality on the training set size, we observed that

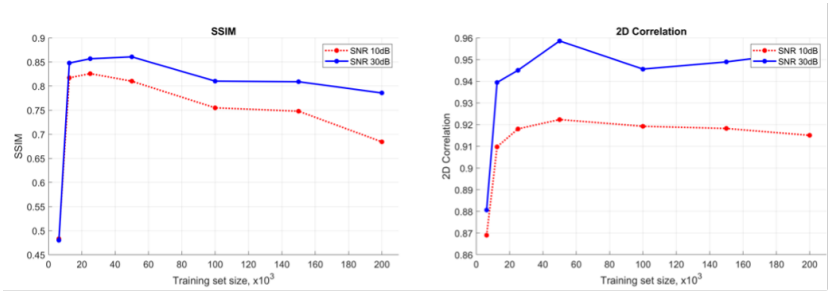


Figure 4: Reconstruction quality metrics dependence on the training dataset size

the best metrics values are achieved when the network is trained on the 50,000 samples. Decreasing the number of samples below about 12,500 causes the metrics' value to drop dramatically, as expected. However, increasing the number of samples also causes a worsening of the reconstruction quality. This behavior requires further investigation. The current working hypothesis is that used learning rates starting values are optimal for the training test size from 12,500 to 50,000 samples, and a bigger training set size requires selecting different values.

We plan to continue our experiments to develop the generic approach of supervised training of cGAN in CCEIT image reconstruction.

References

[1] Y. Jiang and M. Soleimani, "Capacitively Coupled Phase-based Dielectric Spectroscopy Tomography," *Sci. Rep.*, vol. 8, no. 1, p. 17526, 2018.

[2] Y. D. Jiang and M. Soleimani, "Capacitively Coupled Electrical Impedance Tomography for Brain Imaging," *IEEE Trans. Med. Imaging*, vol. 38, no. 9, pp. 2104–2113, 2019.

[3] A. Peyman and C. Gabriel, "Dielectric Properties of Tissues," 2005.

[4] A. Christanto and I. Darinafitri, "The Role of Electrical Impedance Tomography in Lung Imaging," *Cermin Dunia Kedokt.*, vol. 48, no. 9, p. 372, 2021.

[5] S. Y. Qu, M. Dai, S. Wu, Z. R. Lv, X. Y. Ti, and F. Fu, "System introduction and evaluation of the first Chinese chest EIT device for ICU applications," *Sci. Rep.*, vol. 11, no. 1, p. 19273, 2021.

[6] D. S. Holder, "Electrical Impedance Tomography of brain function," *2008 World Autom. Congr. WAC 2008*, vol. 5, no. 2, pp. 87–93, 2008.

[7] B. H. Brown, "Electrical impedance tomography (EIT): a review," *J. Med. Eng. & Technol.*, vol. 27, no. 3, pp. 97–108, 2003.

[8] D. Isaacson and E. L. Isaacson, "Comment on calderón 's paper: "on an inverse boundary value problem," *Math. Comput.*, vol. 52, no. 186, pp. 553–559, 1989.

[9] K. Knudsen, M. Lassas, J. L. Mueller, and S. Siltanen, "D-bar method for electrical impedance tomography with discontinuous conductivities," *SIAM J. Appl. Math.*, vol. 67, no. 3, pp. 893–913, 2007.

[10] C. J. Kotre, "EIT image reconstruction using sensitivity weighted filtered backprojection," *Physiol. Meas.*, vol. 15, no. 2A, 1994.

[11] H. Yan, Y. F. Wang, Y. G. Zhou, and Y. H. Sun, "3D ECT reconstruction by an improved Landweber iteration algorithm," *Flow Meas. Instrum.*, vol. 37, pp. 92–98, 2014.

[12] Y. Li and W. Yang, "Image reconstruction by nonlinear Landweber iteration for complicated distributions," *Meas. Sci. Technol.*, vol. 19, no. 9, 2008.

[13] C. Li, M. Karamehmedović, E. Sherina, and K. Knudsen, "Levenberg–Marquardt Algorithm for Acousto-Electric Tomography based on the Complete Electrode Model," *J. Math. Imaging Vis.*, pp. 492–502, 2021.

[14] Q. Fang, "Distinctions between Levenberg–Marquardt method and Tikhonov regularization," *Dartmouth Coll. Publ.*, pp. 1–7, 2004.

[15] B. McDermott, M. O'Halloran, E. Porter, and A. Santorelli, "Brain haemorrhage detection using a SVM classifier with electrical impedance tomography measurement frames," *PLoS One*, vol. 13, no. 7, Jul. 2018.

[16] T. Rymarczyk, G. Kłosowski, E. Kozłowski, and P. Tchórzewski, "Comparison of Selected Machine Learning Algorithms for Industrial Electrical Tomography.," *Sensors (Basel).*, vol. 19, no. 7, p. 1521, Mar. 2019.

[17] B. Grychtol, B. Müller, and A. Adler, "3D EIT image reconstruction with GREIT," *Physiol. Meas.*, vol. 37, no. 6, pp. 785–800, 2016.

[18] T. Rymarczyk, P. Tchórzewski, and J. Sikora, "Coupling Boundary Element Method With Level Set Method To Solve Inverse Problem," *Informatics Control Meas. Econ. Environ. Prot.*, vol. 7, no. 1, pp. 80–83, 2017.

[19] M. Aller, D. Mera, J. M. Cotos, and S. Villaroya, "Study and comparison of different Machine Learning-based approaches to solve the inverse problem in Electrical Impedance Tomographies," *Neural Comput. Appl.*, vol. 35, no. 7, pp. 5465–5477, 2023.

[20] M. Ivanenko et al., "Image Reconstruction Using Supervised Learning in Wearable Electrical Impedance Tomography of the Thorax," *Sensors*, vol. 23, no. 18, p. 7774, Sep. 2023.

[21] W. Deabes and A. E. Abdel-Hakim, "CGAN-ECT: Tomography Image Reconstruction from Electrical Capacitance Measurements Using CGANs," *arXiv Prepr. arXiv2209.03737*, 2022.

[22] and T. B. Olaf Ronneberger, Philipp Fischer, "U-Net: Convolutional Networks for Biomedical Image," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, no. Cvd, pp. 12–20, 2015.

[23] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 1, pp. 448–456.

[24] J. Kryszyn and W. Smolik, "2D Modelling of a Sensor for Electrical Capacitance Tomography in Ectsim Toolbox," *Informatics Control Meas. Econ. Environ. Prot.*, vol. 7, no. 1, pp. 0–0, 2017.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2016, vol. 2017-Janua, pp. 5967–5976.

Chapter 4

Neural Network and Deep Learning Systems

Track Chairs:

- prof. Aleksander Byrski - AGH University of Ktakow
- prof. Maria Ganzha - Warsaw University of Technology
- prof. Marcin Kurdziel - AGH University of Krakow

Improvements in Vision Graph Neural Networks

Bartłomiej Wójcik[0009−0003−5096−0755]
Arkadiusz Tomczyk[0000−0001−9840−6209]

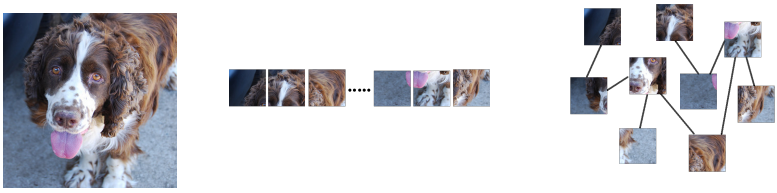
Łódź University of Technology
Institute of Information Technology
al. Politechniki 8, 93-590 Łódź, Poland
bartlomiej.wojcik@dokt.p.lodz.pl
arkadiusz.tomczyk@p.lodz.pl

Abstract. *Vision Graph Neural Networks (ViG) have demonstrated superior performance in computer vision tasks compared to Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs). ViG's adaptability to varying spatial relationships and irregular structures within images, coupled with its dynamic information aggregation, positions it as a robust solution for understanding of both fine-grained details and broader scene context. However, challenges such as vanishing gradient during training and the methods of defining edges need attention. In this work, we propose improvements to ViG, focusing on mitigating vanishing gradient issues, introducing novel edge generation strategies, and incorporating trainable edge weights.*
Keywords: *vision graph meural networks, edge generation, residual connections, adaptive adjacency matrix, graph convolution*

1. Introduction

Vision Graph Neural Networks [1] (ViG) emerge as powerful contenders for computer vision tasks, surpassing Vision Transformers [2] (ViTs) and Convolutional Neural Networks (CNNs) in flexible processing and seamless aggregation of global context. Graph Neural Networks (GNNs), designed to operate on graph-structured data, exhibit a remarkable ability to adapt to varying spatial relationships and irregular structures within images. Their flexibility enables the dynamic aggregation of information across nodes, facilitating effective propagation of context throughout the graph. Unlike the fixed receptive fields of CNNs, GNNs naturally handle complex structures of images. Comapring to ViT, they need not to

DOI:10.17388/WUT.2024.0002.MiNI



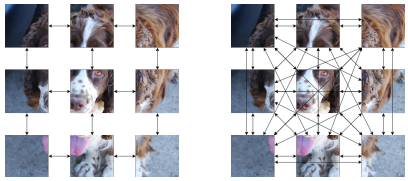
(a) (b) (c)

Figure 1: Comparison of input image representation: (a) - when using CNNs, we are constrained by the grid of pixels in the image, (b) - with the ViT architecture all patches are arranged in a sequence, which is further processed by transformer encoder layers, (c) - ViG allows for arbitrary dependencies to be set between patches.

considered a fully conntected graph and offer a variety of different graph convolutional operators.

The process of graph creation in ViG starts with dividing image into N patches (Figure 1). It is done using a simple CNN block, which transforms each patch into a D dimensional feature vector in the embedding space. All these features are then assigned respectively to a set of unordered nodes V. The next step is the addition of edges E between nodes. Authors of ViG paper generate edges between K nearest neighbours of nodes using the distance calculated in node embedding space. After computing the edges, the graph $G = (V, E)$ is constructed and graph operators are applied. Their goal is to update node embeddings propagating messages along edges from neighboring nodes. The final embeddings are aggregated and passed to a classifier block. Such a model allows for an end-to-end training (botch classical and graph convolutional layers are trained together).

Working with ViGs we have encountered vanishing gradient problem [3]. Backpropagating through these networks causes oversmoothing, eventually leading to features of graph vertices converging to the same value. Moreover, it seems not to be natural to create edges basing on node embeddings. This solution is expensive computationally, as it requires computation of all distances between patches. Additionally, it links only patches with similar embeddings, which means that patches of one object that are visually different will not be connected. Intuitively, however, information about node properties and graph structure should be rather a separate source of knowledge in considered image analysis tasks. In this work, we aim to improve mentioned above ViGs problems.



(a) (b)

Figure 2: Static edge creation: (a) - neighbour edges, (b) - complete edges.

2. Materials and methods

We demonstrate our improvements in effectiveness of ViG model on image classification task. To ensure the comparability of the experiments, in all of them we use similar architecture (Figures 3 and 4): the same CNN block to convert the image into patches, the same number of graph convolutions, global pooling and linear classifier. To compare different architectures we have used the Imagenette[1] dataset, featuring a subset of 10 easily distinguishable classes from ImageNet, containing: tenches, English springers, cassette players, chain saws, churches, French horns, garbage trucks, gas pumps, golf balls, and parachutes. The proposed novelties in ViG's architecture include: alternative static edge creation strategies, residual connections and trainable edge weights.

In contrast to the computationally demanding approach of generating edges based on the K nearest neighbors [1], we propose alternative strategies involving neighbor and complete versions, which provide compelling advantages in the context of graph construction for vision tasks (Figure 2). Rather than fixing on a specific number of neighbors basing on proximity, the generation of neighbor edges offers a more stable solution. Nodes establish connections based on their inherent spatial relationships. Moreover, although it is more expensive computationally, the incorporation of complete edges augments the graph with a global perspective allowing nodes to be linked. This strategy captures long-range dependencies addressing the limitation of the traditional approach, which tends to focus on local relationships. This methods is especially useful for layers using attention, because it allows to assign the weights for each edges and focus only on most significant connections.

[1] `https://github.com/fastai/imagenette`

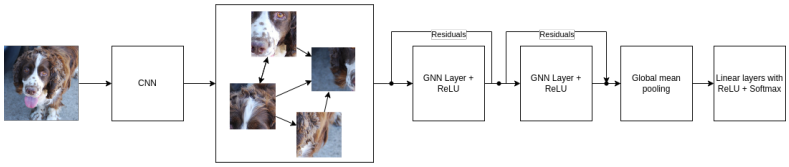


Figure 3: ViG architecture with residual connections.

One of the methods of solving vanishing gradient problem in classical CNNs was the usage of residual connections. It enabled also the creation of deeper architectures. Inspired by this success residual connections were added also in GNNs ([3]). Although, we are not building a deep architecture, in our work we have also used them with graph convolutional layers, which is depicted in Figure 3. This modification of ViG model not only solved the training problems, but it improved model performance as well.

Specification of edges E in a graph G is equivalent with building a graph adjacency matrix A. In this matrix 0 represents no connection between nodes and 1 represents an existing edge. Our next improvement of ViG's architecture assumes that adjacency matrix can be trainable. Its elements can have any value from interval $[0, 1]$ and thus can be treated as edge weights. To achieve that we incorporate an additional CNN block followed by element-wise sigmoid function responsible for dynamic generation of that adjacency matrix. This architecture is presented in Figure 4. To avoid situations where all the elements of A are equal to 0 or 1, we have modified the loss function adding to standard cross-entropy loss a regularization term. This term was equal to $-\lambda \cdot \sigma(A)$ where σ denotes a standard deviation and λ is a regularization coefficient set to 0.25 experimentally. The trained, in this way, edge weights are used by modified graph convolutional layers - every message sent through an edge is multiplied by corresponding weight. Thanks to that we dynamically (depending on the input image) control the influence of different nodes (image patches) on each other.

3. Experiments and results

All our experiments were conducted using three different graph convolutional layers: GraphSAGE (SAmple and aggreGatE) ([4]), Graph Attention Networks (GAT) ([5]) and Graph Transformer ([6]), each model consisted of two GNN lay-

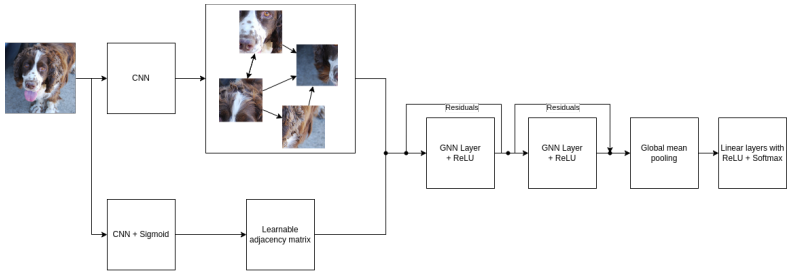


Figure 4: ViG architecture with trainable edge weights.

Table 1: Results of experiments (C - complete edges, N - neighbor edges, R - residuals, TE - trainable edges).

Model	**Accuracy**	**Model**	**Accuracy**
Ours SAGE C-R-TE	0.871 ± 0.003	Ours Trans C	0.848 ± 0.006
Ours Trans C-R-TE	0.867 ± 0.006	Ours GAT N	0.845 ± 0.013
Ours GAT N-R	0.866 ± 0.008	Ours GAT C	0.845 ± 0.009
Ours GAT C-R	0.865 ± 0.009	ViG	0.840 ± 0.004
Ours GAT C-R-TE	0.863 ± 0.005	CNN	0.832 ± 0.021
Ours SAGE C-R	0.860 ± 0.004	Ours Trans N	0.825 ± 0.017
Ours SAGE N-R	0.857 ± 0.012	Ours SAGE N	0.825 ± 0.007
Ours Trans C-R	0.856 ± 0.003	Ours SAGE C	0.822 ± 0.009
Ours Trans N-R	0.852 ± 0.011	ViT	0.746 ± 0.005

ers. ViG and ViT models were reproduced as in the original publications. CNN model consisted of two classic convolutional layers instead of GNN layers. All results are the average of the three trials. Moreover, each group of trials was initialized with the same set of seeds. Every experiment was trained for a maximum of 100 epochs with early stopping on validation accuracy with patience of 20 epochs. Then, the epoch with the best validation accuracy was used for testing. As Imagenette only contains a train and validation dataset, the train dataset of Imagenette was split with fixed seed into train (8500 samples) and validation (969 samples) datasets, and the original validation dataset was used as the test (3925 samples) dataset.

The results presented in Table 1 reveal several notable findings. Firstly, our proposed modifications to the ViG model, particularly those incorporating complete trainable edges (C-R-TE), have led to significant improvements in accuracy compared to traditional convolutional neural networks (CNN) and the Vi-

sion Transformer (ViT) on the Imagenette dataset. This suggests that leveraging graph-based structures and integrating them into convolutional architectures can effectively enhance performance in image classification tasks. Furthermore, the performance of different graph convolutional layers varied, but all the layers indicated the efficacy of this approach in capturing global graph structures for image feature extraction. On the contrary, the ViT model exhibited comparatively lower accuracy on the Imagenette dataset. We hypothesize that this inferior performance may be attributed to the model's reliance on self-attention mechanisms, which may struggle to effectively capture spatial information in smaller datasets like Imagenette. Future investigations on larger datasets will be essential to validate this hypothesis and gain deeper insights into the effectiveness and robustness of different model architectures. In summary, our experiments not only validate the efficacy of graph-based models, particularly those incorporating complete learnable edges, but also highlight the importance of structural information in capturing relationships for image classification tasks. These findings provide valuable insights for the development of more advanced and effective models in computer vision.

4. Summary

Presented results demonstrate the efficacy of proposed modifications to the ViG model, surpassing both traditional CNN and ViT in accuracy on the Imagenette dataset. Graph-based models, particularly those incorporating complete learnable edges, exhibit superior performance, highlighting their potential in image classification tasks. Furthermore, we hypothesize that the inferior performance of the ViT model may be attributed to the small size of the Imagenette dataset. In future investigations, we will validate all methods on larger datasets to gain deeper insights into their effectiveness and robustness.

References

[1] Han, K., Wang, Y., Guo, J., Tang, Y., and Wu, E. Vision gnn: An image is worth graph of nodes. *Advances in Neural Information Processing Systems*, 35:8291–8303, 2022.

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszko-

reit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[3] Li, G., Müller, M., Thabet, A. K., and Ghanem, B. Can GCNs go as deep as CNNs? *CoRR*, abs/1904.03751, 2019. URL http://arxiv.org/abs/1904.03751.

[4] Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216.

[5] Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? *CoRR*, abs/2105.14491, 2021. URL https://arxiv.org/abs/2105.14491.

[6] Shi, Y., Huang, Z., Wang, W., Zhong, H., Feng, S., and Sun, Y. Masked label prediction: Unified massage passing model for semi-supervised classification. *CoRR*, abs/2009.03509, 2020. URL https://arxiv.org/abs/2009.03509.

Voice2Face: Lightweight Neural Lips Movement Generation From Raw Audio

Oskar Bartosz[1], **Łukasz Dąbała**[2]

Warsaw University of Technology
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
oskbsz@gmail.com
Lukasz.Dabala@pw.edu.pl

Abstract. *Algorithmic lip synchronization is dominated by heavy parameterized models such as transformer-based architectures. The goal of this work is to create a model that transforms phoneme representations of human voice into facial visemes using a lightweight model while keeping on high-quality generation.*

Impressive results of generation were achieved on a dataset of real people's voices. The time needed for the generation of a single frame is sufficient for real-time generation (3.38 milliseconds per frame). Generated faces are natural-looking, with each transition between frames performed smoothly. The resulting model weights take up no more than 20KB of disc space.
Keywords: *lip synchronization, lightweight, artificial intelligence*

1. Introduction

Generating lip movement from audio is a complex task requiring expertise in computer vision, graphics, and neural networks. Neural models are growing larger to tackle complex problems, but optimizing memory and time complexity is crucial for efficient computing, especially for real-time applications.

This study draws inspiration from existing solutions, focusing on minimizing complex face representations and enhancing network efficiency for real-time applications. The structure of architecture proposed in this work is most similar to Audio2Face[1]. Here, the issue of rendering from key points is less significant,

DOI:10.17388/WUT.2024.0002.MiNI

as we assume that the crux of the problem lies in the proper alignment of the key points of the mouth to the sound at the best possible time.

The goal of this work is to create a lightweight solution that will enable the creation of a facial representation based on a set of features extracted from raw audio recordings. Such obtained features can be processed further with a backend such as OpenGL or generative AI to obtain talking face frames.

The following chapters provide an introduction to literature and important topics such as computer graphics or signal theory, a description of the program's operation, and the results and conclusions obtained.

2. Related Work

Possible solutions to the facial animation from audio problems vary widely, from classical statistical methods to cutting-edge approaches like transformers [2]. While methods differ, the core idea remains consistent: raw audio is transformed into features to generate the output. Two primary approaches emerge: end-to-end conversion of sound to a human face image [6, 5, 2], and indirect control of facial key points for rendering [1, 7, 4].

While recent models such as Wav2Lip [5] and Meta's Meshtalk [4] achieve impressive realism, they prioritize quality over speed, potentially impacting performance and fps. As facial animation algorithms often integrate with other processing stages, real-time efficiency becomes crucial. Hence, this work emphasizes achieving single-frame inference times suitable for real-time systems amidst the complexity of modern models.

3. Implementation

Facial animation synthesis from voice using AI is a multifaceted challenge. Our approach splits the task in two: Face Autoencoder (FAE) for face representation dimensionality reduction, and Voice2Face (V2F) network for predicting face embeddings from raw audio.

Traditionally, facial animation entails manual input of control bones by programmers to manipulate facial points (e.g., mouth movement, and corner distances). Our adaptation employs an autoencoder to map 2D point positions to a vector of n bones in a hidden space, creating a machine-interpretable representation. Let $\mathcal{K}$ represent normalized key points, comprising flattened x and y co-

ordinates. We derive vector $d \in K$for all mouth key points as described in 1, and vector k for all voice-independent key points.

$$\begin{aligned} &K = \{x_1, y_1, x_2, y_2, \ldots, x_{68}, y_{68}\} \\ &k = [x_1, y_1, x_2, y_2, \ldots, x_{47}, y_{47}] \\ &d = [x_{48}, y_{48}, x_{49}, y_{49}, \ldots, x_{68}, y_{68}] \\ &0 \leq x_i, y_i \leq 1, \quad \text{for } i = 1, 2, \ldots, 68 \end{aligned} \tag{1}$$

$$\begin{aligned} &e = G(d) \\ &d \approx D(G(d)) \\ &e \in \mathbb{R}^5; d \in \mathbb{R}^{20} \end{aligned} \tag{2}$$

The task involves finding a vector e that represents the encoding of d in fewer dimensions, achieved through the FaceAutoEncoder (FAE). The encoder G transforms the vector of mouth keypoints d into face encodings e, while the decoder D performs the inversion of G. Both G and D consist of fully connected layers with ReLU activation. The input and output shapes are tailored to the input face representation, with the first FC layer having an input size of 40. Figure 1 illustrates the FAE architecture, designed to accommodate variations in the length of input key points and encoding vectors across different face landmarks backends.

FAE simplifies face representation by disregarding face key points not correlated with input audio. To estimate face key points from recordings, features are extracted from wave files (W) assigned to videos (V). For each video frame V_i, MFCC spectrograms (m) are generated from voice frames, utilizing Equation 1, with window size determining the number of included frames and the sample rate specifying audio samples per second. Zero padding is employed for initial frames.

$$\begin{aligned} &m_i = MFCC(W[a:b]) \\ &j = i * sample\_rate \\ &a = j - sample\_rate * window\_size \\ &b = j + 1 \end{aligned} \tag{3}$$

$$\begin{aligned} &\hat{e}_0 = e_0 = G(d_{quiet\_face}) \\ &h_i = \hat{e}_{i-1} \\ &\hat{e}_i = F(m_i, h_i) \\ &\text{for } i = 1, 2, \ldots, length(V) \end{aligned} \tag{4}$$

With audio features, face encoding e from Equation 1 can be estimated using features m through the V2F model denoted as F. The network architecture comprises a two-level LSTM and an attention layer. For each m_i corresponding to V_i, $F(m_i, h_i)$ calculates face encoding $\hat{e}_i$. LSTM hidden space h_i is initialized with the previous frame's predicted $\hat{e}_{i-1}$ to ensure prediction continuity. For $i = 1$, $\hat{e}_0$ represents a quiet face, derived using FAE with any quiet face from the training data. Subsequently, frame $\hat{V}_i$ is reconstructed from each approximated $\hat{e}_i$ using the FAE decoder. All key points are obtained by combining voice-dependent key points $\hat{d}$ with voice-independent keypoints k from Equation 2. In equation 5 function *Render* is used to create the resulting image. It can be an OpenGL renderer or a generative AI model.

$$\begin{aligned} \hat{d}_i &= D(\hat{e}_i) \\ \hat{V}_i &= Render(k \cup \hat{d}_i) \end{aligned} \tag{5}$$

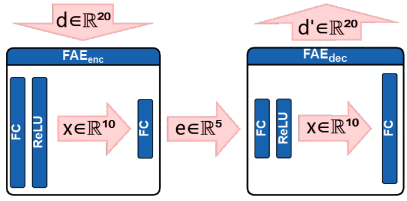


Figure 1: FAE is built only with Fully Connected layers with ReLU as activations.

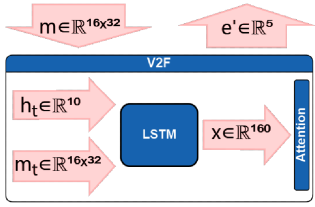


Figure 2: V2F predicts the next face embedding with the use of LSTM and attention layer.

The work centers on creating the training dataset from the **GRID Dataset [9]**, comprising one thousand three-second recordings of faces and voices from thirty-three individuals. Data analysis reveals image issues like varying face sizes and orientations, addressed through preprocessing (face straightening, resizing, and normalization). Face normalization involves aligning faces along a line connecting eye keypoints and adjusting them to the Cartesian coordinate system. Variations in voices are mitigated using MFCC encoding, which condenses audio signal information for speech recognition while reducing feature space dimensionality. Nonstationary noise reduction [8] eliminates constant noise from audio, preventing its influence on facial features in noise-free environments.

4. Conclusion and Results

FaceAutoencoder	
all faces time	35.001ms
single face time	0.350ms
mouth Avg MSE	3.47×10^{-7}
MSE standard deviation	1.81×10^{-7}

Voice2Face	
all frames time	227.997ms
single frame time	3.034ms
mouth Avg MSE	1.23×10^{-4}
MSE standard deviation	6.16×10^{-5}

Table 1: All metrics were measured on a machine with AMD Ryzen 7 6800H CPU and NVIDIA GeForce RTX 3070 Ti GPU. FAE and V2F were evaluated on 75 faces. FAE's error is negligible. V2F processing time is enough to provide real-time calculations. Generated frames highly correspond to ground truth.

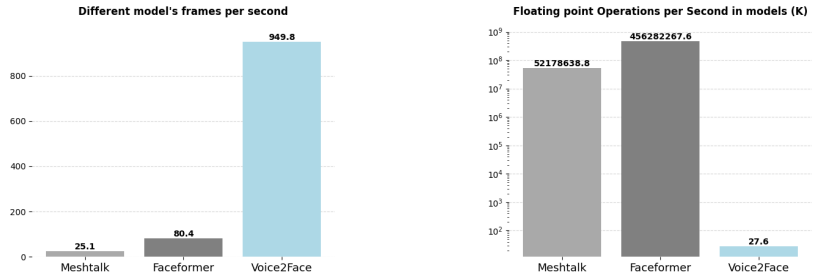


Figure 3: V2F, in comparison to Meshtalk (Meta) and Faceformer, stands out for generating key points without resulting images. Meshtalk employs an intermediate 3D model, while Faceformer generates frames end-to-end. Histograms indicate that V2F offers ample time and parameter resources for further processing, like rendering. Thus, V2F enables rapid key point extraction adaptable to various user needs.

The task has been completed successfully, yielding animations with low latency and high-quality generated visemes. The total frame processing time for FAE and V2M is approximately 3.4 milliseconds, enabling real-time facial movement generation. The proposed models are compact, occupying no more than 20KB of space, suitable even for mobile on-device processing. The project allows for easy scalability by substituting specific action models.

Results from the FAE testing script on one hundred frames indicate low errors, reflecting robust face representation after decoding from latent space (see Table 1). V2F was trained to learn the mapping between phoneme and viseme, achieving good prediction accuracy with compact models (Table 1 illustrates V2F's performance on the GRID test set). Figure 4 compares Wav2Lip and Voice2Face networks, showcasing V2F's comparable key point quality to larger models. Efficiency comparison with Meshtalk and Faceformer is depicted in Figure 3.

References

[1] Guanzhong Tian, Yi Yuan, Yong Liu: AUDIO2FACE: Generating speech/face animation from signle audio with attention-based bidirectional LSTM networks. Arxiv 1905.11142 (2016)

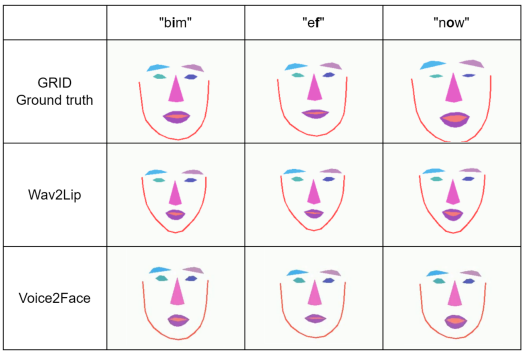


Figure 4: Comparison between Voice2Face and much more complex Wav2Lip[5], on test audio and video from GRID, with random words "bim blue 'f' the seven now". Wav2Lip key points were obtained by using dlib on output images from a model.

[2] Yingruo Fan, Zhaojiang Lin, Jun Saito: FaceFormer: Speech-Driven 3D Facial Animation with Transformers, Arxiv: CVPR Conference (2022)

[3] Community, B. O. (2018). Blender - a 3D modeling and rendering package. Stichting Blender Foundation, Amsterdam. Retrieved from http://www.blender.org

[4] Alexander Richard, Michael Zollhofer, Yandong Wen, Fernando de la Torre, Yaser Sheikh: MeshTalk: 3D Face Animation from Speech using Cross-Modality Disentanglement, Arxiv 2104.08223 (2022)

[5] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, C V Jawahar: A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild, Arxiv 2008.10010 (2020)

[6] Olivia Wiles, A. Sophia Koepke, Andrew Zisserman: X2Face: A network for controlling face generation using images, audio, and pose codes, Arxiv 1807.10550 (2018)

[7] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, Matthias Nießner: Neural Voice Puppetry: Audio-driven Facial Reenactment, Arxiv 1912.05566 (2020)

[8] Tim Sainburg: noisereduce https://doi.org/10.5281/zenodo.3243139 (2019)

[9] Grid Dataset: "https://spandh.dcs.shef.ac.uk//gridcorpus/downloads

[10] Szabolcs Dombi: ModernGL, high performance python bindings for OpenGL 3.3+ (2020)

[11] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, Victor Lempitsky: Few-Shot Adversarial Learning of Realistic Neural Talking Head Models, Arxiv 1905.08233 (2019)

A. Rendering

A simple renderer was created, using Python binding for OpenGL[10]. It works simply by mapping steering key points to face vertices on a 3D model, so face key points condition the output. It is a proof of concept for using the solution in real-time systems. Table 2 shows execution time measurements for processed frames. A speed of less than 10 milliseconds per image is simple to obtain, ensuring real-time. Different output samples are showcased in Figure 5.

Render processing time	
V2F inference	3.051ms
Renderer processing	6.087ms
All processing	9.141ms

Table 2: Measured time for key points generation and rendering.

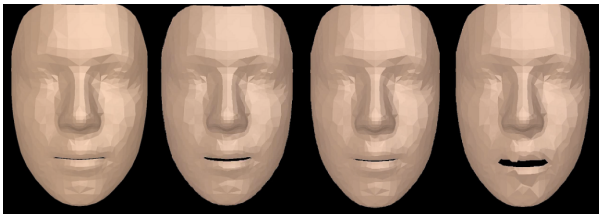


Figure 5: Mapped phonemes, from left: "silence", "too", "si", "ma"

Also, neural rendering techniques could be used, such as rendering with GAN models. As an example, there is a well-understood model with the capability to create frames with very high realism [11]. With the use of this model, one can obtain a video of a given person lip-synced to any voice. Some samples from the original article are shown in Figure 6.

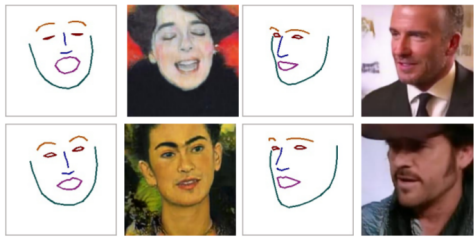


Figure 6: Graphic from article[11]. After retraining the network, it will be possible to use it as a renderer, which is the subject of future work.

Exploration of Metaheuristic Approaches for Tuning Internal Parameters in Probabilistic Neural Networks

Szymon Kucharczyk[1][0009−0002−2413−6984],
Piotr A. Kowalski[2,3][0000−0002−4850−1758]

[1]*AGH Doctoral School, AGH University of Krakow,*
al. A. Mickiewicza 30, 30-059 Krakow, Poland
kucharcz@agh.edu.pl
[2]*Faculty of Physics and Applied Computer Science,*
AGH University of Krakow, al. A. Mickiewicza 30, 30-059 Krakow, Poland
pkowal@agh.edu.pl
[3]*Systems Research Institute, Polish Academy of Sciences*
ul. Newelska 6, 01-447 Warsaw, Poland

Abstract. *This article explores the synergistic relationship between metaheuristic optimisation and the internal parameters of PNNs in the context of pattern recognition and classification tasks. Emphasising the dynamic landscape of metaheuristic procedures, the study aims to enhance PNN learning capabilities through a comprehensive investigation. The primary focus involves meticulous comparative analysis of PNNs utilising diverse kernel functions for classification purposes. The investigation, grounded in well-established datasets, aims to unravel different kernel approaches' inherent strengths and weaknesses. The research concludes with a non-significant impact of kernel selection on PNN training performance when using metaheuristic algorithms.*
Keywords: *probabilistic neural networks, metaheuristics, training, kernel estimators*

1. Introduction

In recent years, the intersection of metaheuristic procedures and neural networks has garnered significant attention within the realm of artificial intelligence

DOI:10.17388/WUT.2024.0002.MiNI

and machine learning. As the field continuously evolves, the development of sophisticated metaheuristic algorithms has become paramount for optimising the performance of neural networks. In particular, Probabilistic Neural Networks (PNNs) have emerged as a powerful paradigm for pattern recognition and classification tasks, offering unique advantages in handling uncertainty. This publication delves into the dynamic landscape of metaheuristic procedures, focusing on their application and refinement to enhance the learning capabilities of PNNs. Through a comprehensive exploration, we aim to elucidate the symbiotic relationship between metaheuristic optimisation and the internal parameters of PNNs, paving the way for advanced and more efficient neural network architectures.

The primary objective of this contribution is to conduct a thorough comparative analysis of PNNs employing various kernel functions for classification tasks. Kernel functions play a pivotal role in shaping the decision boundaries of PNNs, influencing their ability to discern complex patterns within diverse datasets. By systematically evaluating the performance of PNNs with different kernel functions, we aim to unravel the strengths and weaknesses inherent in each approach. The comparison will be grounded in the context of well-known and widely used datasets, ensuring the robustness and generalisability of our findings. This endeavour seeks to provide valuable insights into the nuanced interplay between PNN architectures and diverse kernel functions, ultimately contributing to the advancement of effective and adaptable solutions for classification challenges.

On the other hand, a significant aspect conferring unique strength to PNNs lies in the methodology of nonparametric estimation. This estimation method is largely independent of the shape of the data distribution, rendering the identification and modelling of data more flexible. In the context of PNNs, this characteristic becomes one of the critical elements of the learning procedure, enabling adaptive adjustments to diverse data structures. Analysing this aspect will allow us to comprehend the benefits introduced by various kernel functions in the context of nonparametric estimation and, consequently, how they may impact the classification capabilities of the neural network.

2. Probabilistic Neural Networks

PNNs are memory-based artificial neural networks that are based on the estimation of the probability density function (PDF) using kernel density estimators (KDF) [1]. Due to their architecture, PNNs are especially useful when the dataset

is regularly changed or when class imbalance might be found in the data [2]. Normally, the PNN architecture comprises four layers: input layer, pattern layer, summation layer, and output layer. Initially, the input layer consists of the test pattern features without transformation. The number of neurons in the pattern layer is equivalent to the number of training patterns. Each neuron in the pattern layer calculates the desired Gaussian PDF. The summation layer neurons then aggregate the outputs from the pattern layer for each class category as depicted. Finally, using Bayesian conditional probability, the output layer determines the estimated class for the pattern x.

The key idea of PNN is estimating the probability that the sample x belongs to a given category using kernel estimators. The cumulative probability over all samples is defined as follows [2]:

$$\hat{f}(\mathbf{x}) = \frac{1}{mh_n} \sum_{i=1}^{m} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \tag{1}$$

where m stands the training data cardinality, K is a kernel function, n represents the dimensionality of $\mathbf{x}$, $\mathbf{x}_i$ represents the i-th training pattern, and finally h symbolise Kernel's smoothing parameter. The smoothing parameter is the only trainable parameter in traditional PNNs [3]. It can be established to have the same value for the whole data (data level), different values for each class (class level), different values for each feature (feature level), and different values for each class and feature (class_feature level). This research defines these four cases as *smoothing levels*.

2.1. Kernel estimators

Kernel estimators are a vital part of PNN architecture, affecting the quality of PDF estimation [3]. Probabilistic Neural Networks might be built upon radial or product kernels [2]. In this work, product kernels were used, which implies that for multidimensional datasets, the kernel function must be a product of one-dimensional kernels:

$$K(\mathbf{x}) = K(x_1, x_2, \ldots, x_n)^T = \kappa(x_1) \cdot \kappa(x_2) \cdot \ldots \cdot \kappa(x_n). \tag{2}$$

PNNs can be constructed using various kernel functions, and the choice of kernel function often depends on the specific application. For this study, we opted to use three different kernel functions: *gaussian*, *cauchy* and *laplacian*.

In this research, the *Gaussian*, the *Cauchy* and *Laplacian* kernel functions for the sample vector **x** are defined as follows:

$$\kappa(x_i) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x_i^2}{2}}, \quad \kappa(x_i) = \frac{2}{\pi(x_i^2 + 1)^2}, \quad \kappa(x_i) = \frac{2}{\pi(x_i^2 + 1)^2}. \tag{3}$$

3. Training PNNs with metaheuristics

Metaheuristic methods have been used to train FCNNs [4] and PNNs, including Particle Swarm Optimisation (PSO) [5] and Bat algorithm (BAT) [6]. Generally, PSO and BAT algorithms are iterative processes based on the cooperative behaviour of animals or insects (individuals) in a swarm. They are called particles for the former and bats for the latter. The foundation of swarm-based intelligence algorithms is the ability to share knowledge between particles united in the swarm. This cooperation is assumed to strive towards the optimal solution while passing the infinite variable space of possible solutions.

The swarm's knowledge might be divided into local information and global information. Using metaheuristic methods, leveraging both in a quantified ratio builds a basis for the learning process. For PNN training, each individual was considered a vector of smoothing parameters, representing their position in the solution space. Each of them has its fitness values (defined in section 4), position, and algorithm-specific parameters that might be found in [5], [6]. Indeed, these algorithms process a population of solutions rather than develop one.

Swarm particles were initialised using a random initialisation procedure from range (0, 10). One should note that during the optimisation the h values were restricted to be greater than 0. It is a limitation deriving from the PNN architecture definition [1].

4. Methods and data

Two datasets were used to compare kernel selection impact on PNN classification performance: the Iris dataset [7] and the Breast Cancer dataset [7]. Indeed, the Iris dataset was used as a traditional benchmark for classification performance research. The second one was used to test various PNN kernel additional functions on data with different characteristics (no. of features, class distribution, data domain, data cardinality). With these datasets, the impact of kernel function on a Probabilistic Neural Network parameters optimisation using metaheuristic was

conducted. The PNN training was performed by applying the PSO and BAT algorithms to find sub-optimal smoothing parameters to perform the classification of samples from the mentioned datasets. These swarm algorithms were particularly populated with individuals representing randomly initialised h values, with random weights from range (0, 10) [2]. So, this representation of individuals indicates the structure of the PNN for PSO and BAT methods. Then, these algorithms were run for each dataset, optimising the h parameters for 100 iterations or until the objective function's convergence condition was met. The error rate function (eq. 6) was utilised as a fitness function for metaheuristic training methods. Here, the error rate function highlights the accuracy of the PNN in classifying samples from training and validation datasets. The lower the error rate, the better the accuracy of the Network is assumed. As for the paper's length limitation, a detailed description of applying PSO and BAT for PNN training might be found in [5] and [6], respectively. The metaheuristic procedure parameters used in this study were selected following the referred research. The evaluation and training were performed using a stratified 5-fold cross-validation resampling method to ensure relevant model generalisation [8]. As described in eq. 4 and eq. 5 respectively, accuracy and categorical cross-entropy (log-loss) [8], were calculated to evaluate PNNs classification performance. It is assumed that with these metrics, an estimate of the error rate and the certainty of prediction can be computed, which provides sufficient resources for the model evaluation. Train and test scores from the cross-validation procedure were calculated as a mean value from all folds. Finally, to ensure experiment reproducibility, a random seed was established.

The evaluation metrics and the fitness function are defined as:

$$accuracy(y, \hat{y}) = \frac{1}{m} \sum_{i=0}^{m-1} 1(\hat{y}_i = y_i), \tag{4}$$

$$L_{log}(y_i, p_i) = -(y_i ln(p_i) + (1 - y_i) ln(1 - p_i)), \tag{5}$$

$$error\ rate = 1 - \frac{number\ of\ correct\ predictions}{cardinality\ of\ test\ sample}, \tag{6}$$

where y stands for class labels from the data, p and $\hat{y}$ denote the class probability and labels predicted by the model, respectively.

5. Results and conclusions

Here, a comparison of kernel function impact on PNN classification performance trained with PSO and BAT algorithms is presented. The computations were conducted for four smoothing parameter levels. Two well-established UCI datasets were used to compare the PNN performance. Table 1 highlights the outcomes of the PNN training using the PSO method for the Iris dataset. Similar tables were produced for other datasets and training methods but are not presented here due to text length limitations. In addition, Tables 2 - 5 present calculations of standard deviation (σ) between different kernels used for PNN classification training.

Tables 4 and 5 show that the kernel selection has a negligibly small impact on the PNN training performance when training them with metaheuristics (specifically PSO and BAT procedures). The standard deviation for the Breast Cancer dataset was approximately around 0 for both training and test metrics values, with singular exceptions for the data smoothing level.

Similarly, the Iris dataset's standard deviation of performance metrics was also relatively small but more significant than for the Breas Cancer datasets. Tables 2 and 3 highlight the standard deviation for the Iris dataset, which is approximately 1 to 2 orders of magnitude lower than the value of each metric itself.

In conclusion, based on the outcomes of the research performed, it is implied that kernel selection has no significant impact on PNN training with metaheuristics. This result is assumed to align with PNN and metaheuristics' flexibility for nonparametric estimation. Furthermore, it reflects these methods' ability to adapt to various data structures and representations, e.g., by applying different PNN kernel functions. In the future, this study might be extended to include more datasets and a detailed analysis of the impact of smoothing levels on training performance.

Acknowledgment

This work was partially supported by the program „Excellence Initiative – Research University" for the AGH University of Krakow and by a Grant for Statutory Activity from the Faculty of Physics and Applied Computer Science of the AGH University of Krakow.

Table 1. Comparison of PNN training performance for different kernels and smoothing parameter level for the Iris dataset and PSO optimisation method.

Smoothing Level	Kernel	accuracy_train	accuracy_test	log_loss_train	log_loss_test
class	gaussian	0.777	0.733	1.224	1.224
	cauchy	0.930	0.907	0.629	0.622
	laplacian	0.883	0.860	0.994	1.000
class_feature_matrix	gaussian	0.915	0.927	0.895	0.880
	cauchy	0.915	0.920	0.766	0.765
	laplacian	0.950	0.907	0.797	0.802
data	gaussian	0.950	0.927	0.66	0.658
	cauchy	0.942	0.947	0.530	0.561
	laplacian	0.950	0.953	0.676	0.688
feature	gaussian	0.948	0.933	0.970	0.973
	cauchy	0.942	0.947	0.489	0.492
	laplacian	0.958	0.960	0.574	0.558

Table 2. PNN performance metrics σ comparison between kernels, for various h levels for the Iris dataset and PSO optimisation method.

Smoothing Level	accuracy_train	accuracy_test	log_loss_train	log_loss_test
class	0.079	0.090	0.300	0.304
class_feature_matrix	0.020	0.010	0.067	0.059
data	0.005	0.014	0.080	0.066
feature	0.008	0.013	0.257	0.261

Table 3. PNN performance metrics σ comparison between kernels, for various h levels for the Iris dataset and BAT optimisation method.

Smoothing Level	accuracy_train	accuracy_test	log_loss_train	log_loss_test
class	0.115	0.080	1.401	0.587
class_feature_matrix	0.034	0.045	0.106	0.084
data	0.008	0.070	0.017	0.161
feature	0.008	0.032	0.094	0.080

Table 4. PNN performance metrics σ comparison between kernels, for various h levels for the Breast Cancer dataset and PSO optimisation method.

Smoothing Level	accuracy_train	accuracy_test	log_loss_train	log_loss_test
class	0.001	0.013	0.000	0.000
class_feature_matrix	0.001	0.003	0.000	0.000
data	0.001	0.015	2.616	0.095
feature	0.005	0.052	0.000	0.000

Table 5. PNN performance metrics σ comparison between kernels, for various h levels for the Breast Cancer dataset and BAT optimisation method.

Smoothing Level	accuracy_train	accuracy_test	log_loss_train	log_loss_test
class	0.000	0.039	0.040	0.0
class_feature_matrix	0.000	0.044	0.000	0.0
data	0.000	0.067	2.836	0.0
feature	0.001	0.031	0.000	0.0

References

[1] Specht. Probabilistic neural networks for classification, mapping, or associative memory. In *IEEE 1988 International Conference on Neural Networks*, pages 525–532 vol.1. 1988.

[2] Kowalski, P. A., Kusy, M., Kubasiak, S., and Łukasik, S. Probabilistic neural network - parameters adjustment in classification task. pages 1–8. 2020. doi: 10.1109/IJCNN48605.2020.9207361.

[3] Specht, D. Enhancements to probabilistic neural networks. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 761–768 vol.1. 1992. doi:10.1109/IJCNN.1992.287095.

[4] Kowalski, P. and Łukasik S. Training neural networks with krill herd algorithm. *Neural Process Lett*, 44:5–17, 2016. doi:10.1007/s11063-015-9463-0.

[5] Ciarelli, P., Krohling, R., and Oliveira, E. *Particle Swarm Optimization Applied to Parameters Learning of Probabilistic Neural Networks for Classification of Economic Activities*, chapter 19, pages 313–328. 2009.

[6] Naik S.M., K. V., Jagannath R.P.K. Bat algorithm-based weighted laplacian probabilistic neural network. *Neural Comput & Applic*, 32:1157–1171, 2020. doi:10.1007/s00521-019-04475-4.

[7] Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[8] Pedregosa, e. a. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Predicting churn with RNN based neural networks models in e-book subscription domain

Drozda Paweł[1,2][0000−0003−3163−9408],
Ropiak Krzysztof[1,2][0000−0001−8314−0276],
Mozalewski Łukasz[2], **Mikołaj Małaczyński**[2],
Mateusz Frukacz[2]

[1]*University of Warmia and Mazury in Olsztyn, Poland*
{pdrozda,kropiak}@matman.uwm.edu.pl
[2]*Legimi JSC*
{lukasz.mozalewski, mikolaj.malaczynski, mateusz.frukacz}@legimi.com

Abstract. *Churn prediction is one of the key areas of interest for companies offering subscription-based products. The search for effective methods to predict service abandonment and the characteristics that influence is of interest to managers and scientists. This paper deals with the problem of churn in e-book subscription domain using recurrent neural networks (RNNs) for Legimi JSC. The application of RNNs allowed to achieve satisfactory results, where in the best case the value of the parameter F1 reached 90%.*
Keywords: *churn, RNN, bag of activities, time vector, e-book domain*

1. Introduction

The motivation for using recurrent networks to develop an effective churn prediction model was its ability to retain information from previous iterations through the network, while also being capable of accommodating sequences of varying lengths to address diverse business requirements. The problem of customer churn has been widely addressed in research in various areas due to the fact that in many cases the growth of companies depends on customer loyalty. Therefore, predicting future abandonment and trying to prevent it is often a key part of a company's efforts.

DOI:10.17388/WUT.2024.0002.MiNI

One area where churn research is in high demand is telecom. RRNs have been used in [1] to make predictions based on data such as demographics, operator network usage and other user account data. Also in [2] a similar approach was taken. Another area of research was presented in [3], where authors present experiments on churn prediction for users of an online card game. The authors first reduced the churn prediction problem to a regression problem, where the model estimates the time it takes for a user to remain logged out based on behavioral data.

This paper is the first which considers customer churn in the e-book subscription domain. For experiments, three versions of recurrent neural networks have been applied: Vanilla RNN[4], LSTM[5], GRU[6] and for two approaches in terms of data preparation: Bag of Activities and Time Vector. This solution proved to be successful, where for the Bag of Activities approach, all versions of the RNNs reached 90% for the F1 parameter, which could potentially be used in a production environment.

2. Methodology

A characteristic of recurrent networks is that in the next pass, some part of the activation values calculated for the previous position of the sequence is retained, which acts as a kind of memory that favors sequences closer to the one currently being processed, but does not "forget" completely the values calculated earlier.

Three recurrent network architectures were used in the study: Vanilla RNN (classical recurrent networks), LSTM (Long-Short Term Memory) and GRU (Gated Recurrent Unit).

The experiments have been divided into two different approaches, which differ in both the choice of dataset features and the final construction of the classification model. For simplicity of identification in this article, we will call them model BOA and the Time Vector model.

Description of the BOA model

The name of the model was adopted from a methodology similar to that known from NLP (natural language processing) called Bag Of Words[7], but instead of words it is an aggregation of user events monitored in an existing system. This is a classification model based on recursive networks that, based on a matrix of user reading characteristics, classifies the user into one of two groups - churn and non-churn. This model, unlike the next approach, consists of a single recursive

network.

Description of the time vector model

This is a hybrid approach consisting of two models. Both are based on recurrent networks, but the first is a regression model to predict the values of a user's reading characteristics in subsequent time periods, and the second is a classification model that, based on data for previous periods and data from the regression model, classifies a user as churn or non-churn. This approach allows not only to predict the binary value of churn or non-churn, but also to estimate the future value of reading characteristics. This model can be used to predict churn in subsequent periods by applying a sliding window on the input data from subsequent regression model predictions. However, this approach was abandoned at the stage of selecting a model more suited to the company's needs. A visualisation for a single forward period is presented below.

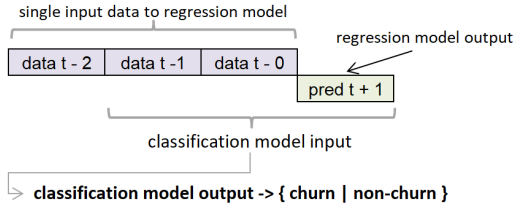


Figure 1. Simplified visualization of the data flow between two time vector models.

3. Dataset preparation

BOA model

For the recurrent neural networks, the Bag Of Activities (BOA) dataset was transformed into a time series, i.e. from flat vectors to matrices, where the rows are the months sorted in descending order and the columns are the Legimi user activity features.

The features for the recursive networks were selected from the bag-of-activities dataset for the 12-month granularity using the lasso regression[8] method (also known as L1 regularization).

Final dataset, with carefully selected features, contains information on the time

spent reading, the number of books read and unfinished, as well as more detailed information on each user's reading sessions, which remains classified information protected by Legimi.

Number of Legimi users selected for the dataset is as follows: churn users - 80014, non-churn users - 86892.

Each user is represented by 12 features in 12 time periods, giving a 12x12 matrix of features arranged chronologically plus a decision parameter.

	Date	user_1	…	user_n
time ↓	period_t1	[feature_1, …, feature_u]	…	[feature_1, …, feature_u]
	…			
	period_tk	[feature_1, …, feature_u]		[feature_1, …, feature_u]

Figure 2. Visualization of data vector for the model BOA.

Where n is the number of users in the dataset, k the number of time periods, u the number of features for each user.

Time vector model

In this approach, 11 features were extracted at the feature selection stage (plus a decision parameter) for each time period, and then each series was transformed into a flat vector starting from the most distant time periods to the most recent values derived from the regression model prediction. This operation results in a single vector for each user.

The model uses information on a selection of 182301 users from the Legimi database with the following distribution: training set: 70 % (127610 users), test set: 20 % (36460 users) and validation set: 10 % (18231 users).

time →

user	f_1_time - k	…	f_u_time - k	f_1_time - 0	…	f_u_time - 0
user_1	value_t - k	…	value_t - k	value_t - 0	…	value_t - 0
…	…	…	…	…	…	…
user_n	value_t - k	…	value_t - k	value_t - 0	…	value_t - 0

Figure 3. Visualization of data vector for the time vector model.

4. Experiments and results

In validating the quality of the models, the experiments were carried out using the CV5 cross-validation method, which means randomly dividing the set into 5 parts, one of which becomes the test set (in the time vector model, the part goes into the validation set) and the other 4 are combined into the training set.

The BOA model experiments used different time periods for data granulation, but the model performed best for longer periods, so only those results are presented below.

Table 1. Cross validated classification results for model BOA.

Model	**CV prec.**	**CV recall**	**CV f1**	**CV f1 macro**
Vanilla RNN	0.91	0.88	0.90	0.90
LSTM	0.90	0.88	0.89	0.90
GRU	0.90	0.90	0.90	0.90

The main objective of the Time vector is to maximize the classification quality of the RNN-based model and correctly predict whether a user is a churn or not, so the regression model results are limited only to the mean squared error measure presented below.

- Vanilla RNN: training loss MSE = 0.286, validation loss MSE = 0.646
- GRU: training loss MSE = 0.588, validation loss MSE = 0.629
- LSTM: training loss MSE = 0.449, validation loss MSE = 0.650

The experiments carried out using the Vanilla RNN, LSTM and GRU architectures yielded very similar results, the differences in results being within statistical error and for this reason only summary tables with averaged results will be presented below.

Table 2. Cross validated classification results for time vector model.

	precision	**recall**	**f1-score**	**support**
churn	0.79	0.96	0.86	15857
non-churn	0.96	0.80	0.87	20603

Table 3. Cross validated classification results for time vector model.

macro avg precision	**macro avg recall**	**macro avg f1**	**accuracy**	**support**
0.87	0.88	0.87	0.87	36460

5. Conclusion

The results of the experiments in the context of classification measures give good results that can be applied to production solutions. A key issue is the correct preparation of the user data, where the most important issue seems to be the definition of the boundary between active and inactive users, which will not be defined in a zero-one manner in the context of the characteristics of their activity in the application. In the data studied, slightly better results were achieved for data from more periods than for users with a shorter activity history. The use of the hybrid time vector model did not provide any additional benefits compared to the single RNN model.

Acknowledgments

The work is part of the project No POIR.01.01.01-00-1970/20, titled "Development of models of behavioral analysis of users using artificial intelligence algorithms to predict consumer behavior, product recommendations and automate marketing and sales processes in the book industry.", which is co-financed by the European Union from the European Regional Development Fund under the Smart Growth Operational Program 2014-2020, Measure 1.1: R&D projects of enterprises, Sub-measure 1.1.1: Industrial research and development work carried out by enterprises. The project is carried out as part of the competition of the National Center for Research and Development 6/1.1.1/2020 - Fast Track.

References

[1] Sudharsan, R. and Ganesh, E. N. A swish rnn based customer churn prediction for the telecom industry with a novel feature selection strategy. *Connection Science*, 34(1):1855–1876, 2022.

[2] Hu, J., Zhuang, Y., Yang, J., Lei, L., Huang, M., Zhu, R., and Dong, S. prnn: A recurrent neural network based approach for customer churn prediction in

telecommunication sector. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4081–4085. 2018.

[3] Xi, M., Luo, Z., Wang, N., Tao, J., Li, Y., and Yin, J. A latent feelings-aware rnn model for user churn prediction with only behaviour data. In *2020 IEEE International Conference on Smart Data Services (SMDS)*, pages 26–35. 2020.

[4] Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–8, 1982. doi:10.1073/pnas.79.8.2554.

[5] Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

[6] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[7] Harris, Z. S. Distributional structure. *WORD*, 10(2-3):146–162, 1954. doi:10.1080/00437956.1954.11659520.

[8] Hans, C. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.

Comparative Study on Different Hyperdimensional Sequence Aggregation Strategies in Dynamic Gesture Recognition

Kacper Kubicki[0000−0002−1811−8395], **Piotr Łuczak**[0000−0002−2530−0283], **Jacek Kucharski**[0000−0002−8704−1950] **Krzysztof Ślot**[0000−0003−1228−0970],

Lodz University of Technology
Institute of Applied Computer Science
Stefanowskiego 18, 93-537 Łódź, Poland
kacper.kubicki@p.lodz.pl

Abstract. *In this work, we investigate the impact of different hyperdimensional sequence aggregation strategies on the performance of the Hybrid, neuro-symbolic classification model. The three considered strategies: permute-and-sum and permute-then-sum with two different tie-breaking ways, differ both in principles of individual sequence element aggregation as well as in the amount of noise introduced. We tested these methods on a hand gesture classification task, based on sequences of images captured with a dynamic vision sensor. The results show superiority of the permute-then-sum aggregation with random-tie-breaking in terms of the classification accuracy, which, however, comes at a cost of increased computational and memory requirements.*

Keywords: *artificial intelligence, hyperdimensional computing, neuromorphic architectures, dynamic vision sensors*

1. Introduction

Encoding information extracted from real-world data such as images or image sequences onto high-dimensional binary or bipolar vectors (HVs) is a challenging problem in hyperdimensional computing (HDC) [1]. One of the key components of HDC that enables the aggregation of multiple HVs into a single one is the hyperdimensional bundling operator. However, different implementations of this operator

may have different effects on the accuracy and robustness of the computational architectures utilizing aggregated HVs. This paper extends our previous work [2] by repurposing the proposed hybrid neuromorphic computational architecture to classify hand gestures. We also present a comparative study of three different sequences aggregation methods, which shows that the highest accuracy for the considered task and for the adopted specific architecture, is offered by random tie-resolution applied to permute-then-sum bundling scheme.

2. Related work

HDC paradigm introduced by Pentti Kanerva [1] is a biologically inspired computational concept, which attempts to model brain activity pattern processing by leveraging high-dimensional (with the dimension d at the order of thousands or more) binary or bipolar vectors. Several advantages, including extreme noise robustness and simplicity of data processing, consisting of local, bitwise operations on binary variables, make HDC an exceptionally promising candidate for energy-efficient and high-performance tasks. HDC has been successfully applied to solve a variety of problems belonging to areas as wide as robotic control, DNA string matching, classification of texts, acoustic signal recognition, analogical reasoning, cognitive modeling and many others [3].

At its core, HDC is built upon a set of surprisingly simple yet efficient operations, namely bundling, binding and permutations. Bundling, equivalent to the pointwise addition, is denoted by $A + B$ and yields a vector that is similar to the argument vectors. Binding, equivalent to the pointwise multiplication, is denoted by $A * B$ and yields a vector that is dissimilar to the argument vectors. Finally, permutation, denoted by Π, rotates the coordinates of a HV and is typically implemented by cyclic shift-by-one [4].

3. Methods

The aggregation of a sequence of HVs, a.k.a. *flattening* or *leveling* the sequence, can be achieved through two distinct methods: permute-and-sum (PaS) and permute-then-sum (PtS) [1]. Given a sequence denoted as ABCD, the permute-and-sum method can be represented as:

$$S = [\Pi[\Pi[\Pi A + B] + C] + D] \tag{1}$$

while permute-then-sum takes the form:

$$S = [\Pi\Pi\Pi A + \Pi\Pi B + \Pi C + D] \tag{2}$$

where Π denotes a permutation operator, + denotes a bundling operator and brackets indicate a normalization of sum vector to $\{-1, +1\}^d$, with a mechanism to brake the ties, necessary in the case of an even-input sequence. Although tie-breaking can be accomplished in several different ways, such as at random (by bundling an additional random HV into the record) [1] or by creating an additional feature (e.g., by binding any two input HVs and bundling the resulting HV into the record) [5], it is nevertheless a source of noise, which is magnified whenever a tie-breaking is needed.

The contributions of this paper are two-fold. First, we show that one can successfully apply the hybrid neuromorphic algorithm that combines convolutional neural network with a HDC module, presented in [2], for dynamic gesture recognition, offering a lightweight solution. Secondly, we compare impact of the choice of a particular sequence aggregation strategy on gesture recognition accuracy.

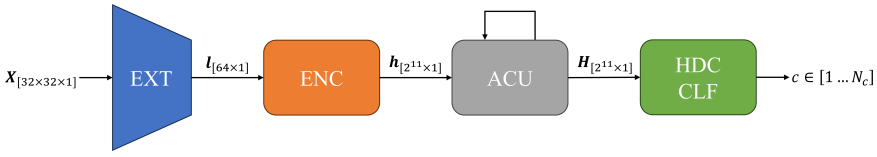


Figure 1. A hybrid, neuro-symbolic architecture proposed in [2]. The architecture consist of a convolutional extractor EXT, encoder ENC projecting a latent representation onto the HVs, an accumulator ACU which aggregates the sequences and a hyperdimensional classifier HDC CLF which selects a winning class based on the assessment of the proximity (calculated using Hamming distance) of the analyzed sequence to the set of known HVs derived for the considered classes.

4. Results

The task selected for the neuro-symbolic architecture proposed in our previous work [2] and presented in Figure 1 was dynamic gesture recognition. We utilize a custom self-collected dataset comprising a total of 980 recordings of 7 different hand gestures taken under 7 different lighting conditions, with each gesture repeated

20 times. The data has been collected using a dynamic vision sensor a.k.a an event camera - a bio-inspired sensor that asynchronously captures per-pixel brightness changes. An event camera encodes these changes using ternary data (-1 for decreased intensity, 0 for no change, $+1$ for increased intensity). Its advantages, such as low latency, high dynamic range and low power-consumption make it suitable for scenarios challenging for traditional cameras [6]. An example of a sequence of twelve pseudo-frames (created by aggregating an event stream using 250 ms time windows) that have been used an input to the convolutional extractor can be seen in Figure 2.

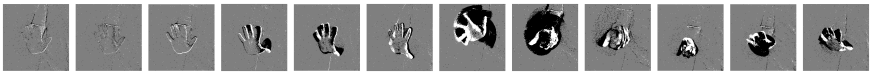


Figure 2. Sequence of twelve pseudo-frames capturing a "clockwise" gesture recorded using an event camera.

We checked three methods of aggregating the sequences of input image representations, mapped onto the HVs: permute-and-sum, permute-then-sum with tie-breaks resolved by bundling an additional feature, and permute-then-sum with tie-breaks resolved at random. The results of classification of sequences into one of the seven gesture categories, presented as a mean J score with 95% confidence intervals (we used leave-one-out cross-validation strategy by treating recordings taken under specific lighting conditions as a testing set), as well as the result of classification of the individual images (without aggregating them into a sequence) by conventional MLP, can be seen in Figure 3.

As depicted in Figure 3, the permute-then-sum strategy, regardless of the tie-breaking procedure, outperforms the permute-and-sum method. This allows the hybrid computational architecture to exceed the classification accuracy of a conventional MLP.

5. Conclusions

The results demonstrate that the permute-then-sum technique for flattening HVs sequence enhances classification accuracy by reducing the occurrence of tie-breaking situations and thereby minimizing the noise introduced into the system. However, the trade-off is the increased memory usage due to the need for separate

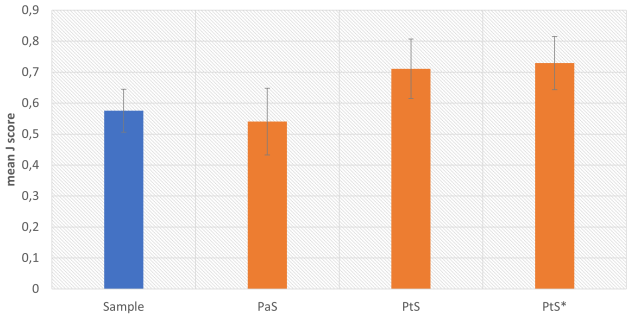


Figure 3. Classification performance for different sequence aggregation methods: permute-and-sum (PaS), permute-then-sum with tie-breaks resolved by bundling an additional feature vector (PtS) and permute-then-sum with tie-breaks resolved at random (PtS*). "Sample" denotes the results of classification of individual images using conventional MLP.

storage of the intermediate results (permuted HVs for each sequence frame). Therefore, a choice of a particular strategy depends on the specific requirements of the application, such as the available memory and the tolerance for noise.

Acknowledgment

This work was funded by European Union's Horizon 2020 research and innovation programme under grant agreement no 101016734. This work has been completed while the 1st author was the Doctoral Candidate in the Interdisciplinary Doctoral School at the Lodz University of Technology, Poland.

References

[1] Kanerva, P. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computation*, 1(2), 2009. doi:10.1007/s12559-009-9009-8.

[2] Łuczak, P., Ślot, K., and Kucharski, J. Combining Deep Convolutional Feature Extraction with Hyperdimensional Computing for Visual Object Recognition.

In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. 2022. doi:10.1109/IJCNN55064.2022.9892281.

[3] Kleyko, D., Rachkovskij, D., Osipov, E., and Rahimi, A. A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part II: Applications, Cognitive Models, and Challenges. *ACM Comput. Surv.*, 55(9), 2023. ISSN 0360-0300. doi:10.1145/3558000. URL `https://doi.org/10.1145/3558000`.

[4] Rahimi, A., Kanerva, P., Benini, L., and Rabaey, J. M. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of exg signals. *Proceedings of the IEEE*, 107(1):123–143, 2019. doi:10.1109/JPROC.2018.2871163.

[5] Schmuck, M., Benini, L., and Rahimi, A. Hardware Optimizations of Dense Binary Hyperdimensional Computing: Rematerialization of Hypervectors, Binarized Bundling, and Combinational Associative Memory. *J. Emerg. Technol. Comput. Syst.*, 15(4), 2019. ISSN 1550-4832. doi:10.1145/3314326. URL `https://doi.org/10.1145/3314326`.

[6] Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi:10.1109/TPAMI.2020.3008413.

Enhancing Dispersed Data Classification: A Hierarchical Model Based on Neural Networks

Kwabena Frimpong Marfo[1][0000−0003−2226−9097]
Małgorzata Przybyła-Kasperek[1][0000−0003−0616−9694]

[1] *University of Silesia in Katowice*
Institute of Computer Science,
Będzińska 39, 41-200 Sosnowiec, Poland
kwabena.marfo@us.edu.pl, malgorzata.przybyla-kasperek@us.edu.pl

Abstract. *The paper uses dispersed data, that is, data collected from independent sources. More specifically, dispersed tabular data is considered with the assumption that a set of local decision tables are available. In as much as each local table has attributes and objects unique to it, there may be attributes and objects that are common in more than a local table. The proposed approach uses a hierarchical model – at the lower level, local tables are transformed using a feature extraction method to generate tables with homogeneous structure. At the higher level, a neural network model is created based on such transformed local tables. Some advantages that the model presents are data protection (since raw data are not sent to the global model), the use of a global model despite the use of dispersed data and a reduced feature space – reducing model complexity. The paper also describes a classification method of new cases based on such a hierarchical model.*
Keywords: *Dispersed Data, Neural Networks, Principal Component Analysis, Multilayer Perceptron*

1. Introduction

The increasing volume and complexity of data in various domains and the limitations of centralized learning models resulted in exploration of dispersed data and dispersed learning paradigms. Perceptions on the use of dispersed data vary depending on context and approach.

Distributed learning involves training models across multiple nodes, each processing a subset of the data. This approach is particularly valuable in scenarios

where centralized processing is impractical due to the sheer volume of data or the need for real-time learning. Distributed learning algorithms have been explored to improve model training efficiency and scalability. Some of the well-known and effective approaches are: XGBoost [1], LightGBM [2], Random Forest [3] among others.

The approach discussed in this paper is also related to distributed learning. We assume that dispersed data collected independently in the form of local tables are available. Moreover, local tables do not satisfy any attribute or object constraints. This means that sets of attributes may be different between local tables, but some attributes may be common. In addition, the number of attributes between local tables can vary. Initially, the use of the k-nearest neighbor classifier as a base classifier for each local table separately was analyzed, and then the obtained prediction results were aggregated using neural networks. It should be mentioned that different types of neural networks were considered [4, 5, 6]. Next, an approach was analyzed in which neural networks with the same structure were built locally, using artificial objects. Such networks were then aggregated by averaging [7]. A completely different approach is now being considered, which aims to generate completely different local tables with an equal number of attributes based on the original tables. For this purpose, different approaches can be used as a transformation of the original data using a reduced number of components. Here we mean methods such as Principal Component Analysis (PCA) [8], autoencoders, Linear Discriminant Analysis (LDA) [9] or Singular Value Decomposition (SVD) Such newly generated tables are then aggregated with an additional attribute that stores the table's identifier. Based on such data, a single neural network model is built. The proposed system is hierarchical. At the lower level are models that perform transformation and reduction of attributes. On the other hand, at a higher level, a neural network model is available. Classifying a new object also goes through two levels of hierarchy. Preliminary experiments show that the use of this approach is promising. This approach guarantees the protection of the data, as the raw data is not shared. It should also be noted that a single neural network model is ultimately obtained.

Federated Learning is a decentralized machine learning approach where models are trained collaboratively across multiple devices or nodes without centralizing raw data [10]. This collaborative training process aims to preserve data privacy, making it particularly relevant when sensitive information is involved. Therefore, the approach presented in the paper is related to federated learning. In horizontal Federated Learning [11], each node possesses a subset of the feature space but

shares the same label space. Vertical Federated Learning [12], on the other hand, deals with vertically partitioned data, where different nodes hold complementary information about the same set of instances. In the presented approach, the data is partitioned both vertically and horizontally. Federated Learning often involves heterogeneous and non-identically distributed (non-IID) data across nodes [13]. Addressing the challenges of non-IID data distribution is crucial for achieving effective model convergence and performance. Federated Learning is often combined with differential privacy techniques to enhance privacy guarantees. In this paper, we put less emphasis on data protection and more on classification quality. In addition, federated learning usually has an iterative form based on convergence [14]; here, the model is built only once.

The rest of the paper is organized as follows. In Section 2, we present the details of the model. We also describe the method to classify the test object. The last section is a conclusion and some future plans and the challenges.

2. Methods and models

The approach proposed consists of two steps. In the first step, based on each local table a modified table with an equal number of attributes in each table specified as parameter k is designated. In the second stage we train a global neural network model based on the concatenated modified local tables.

The first step transforms each local table by use of a feature extraction method to obtain resulting tables with homogeneous number of attributes. Suppose a set of n decision tables $D_i = (U_i, A_i, d)$, $i = 1, \ldots, n$ from one discipline is available, where U_i is the universe, a set of objects; A_i is a set of conditional attributes; d is a decision attribute. With a predetermined number of features k, a transformation map T_i^k is built to transform each D_i to a matrix of vector of principal component scores M_i^k. The transformation map converts the vector of values of the object on the original conditional attributes into a completely different vector of values with a different dimension equal to k; each $\mathbf{t} \in M_i^k$ is a vector of values $\mathbf{t} = (t_1, \ldots, t_k)$. Transformation map is not shared among local units during the creation of the central model, for this reason the raw data is protected. Moreover, both the information about the number of attributes present in each original table and what those attributes are is not shared. As was mentioned earlier, for each local table these data can be different. Thus, it is rather impossible to reconstruct the original form of local tables due to the very large number of unknown parameters. Any feature

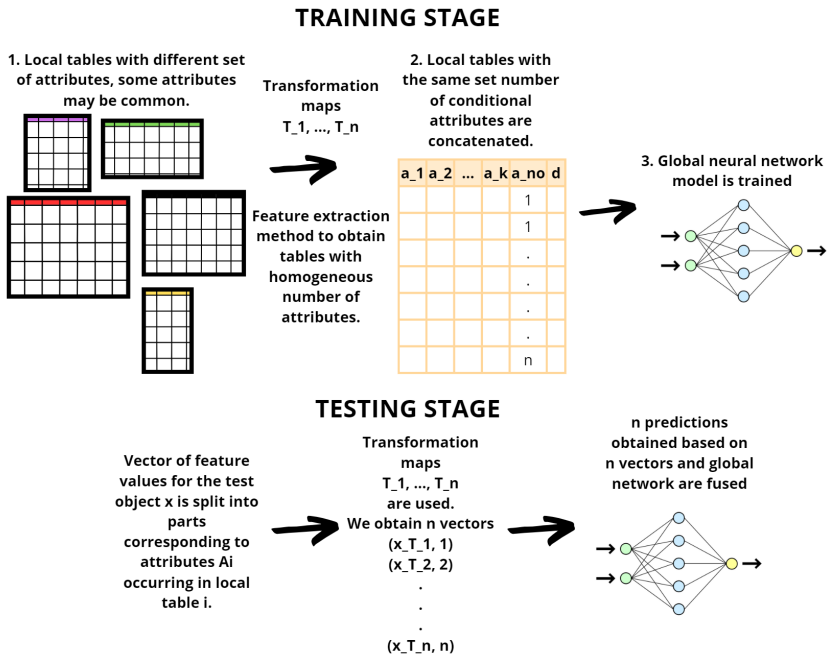
extraction approach such as PCA, LDA or SVD can be used in this step.

In the second stage, all the matrices M_i^k are concatenated (since they have the same number of coordinates) with an additional attribute that stores the table's identifier and label. Concatenated matrix M is used to build a neural network model – second step global model. Different network architectures can be used – from the Multilayer Perceprton (MLP) to Radial Basis Function Networks or Convolutional Neural Networks.

The process of classifying new cases has also two stages. The vector of feature values for the test object is split into parts – corresponding to attributes A_i occurring in local table i. Then the part $\mathbf{x}_i = (a_1(\mathbf{x}), \ldots, a_m(\mathbf{x}))$, where $A_i = \{a_1, \ldots, a_m\}$ are sent to local unit i. The local unit i use the previously determined transformation map T_i^k to obtained $k-$dimensional vector for the test object $\mathbf{x}_{T_i^k}$. The transformed values, along with an additional coefficient denoting the table's identifier are then used as input to the global neural network and prediction is made. In the case where all local tables participate in the classification, we get n predictions for one test object. To obtain the final result, a fusion method such as the sum rule, product rule, the Borda count method, or any kind of voting can used to aggregate the obtained predictions. In this paper, the soft voting used as the fusion method. Figure 1 shows the stages of building the global model and test objects classification process.

In this paper, due to limited space, we only present preliminary experimental results for the approach described above. During experiments, the linear kernel PCA in the first stage and the MLP with a hidden layer in the second stage were used. The tested model's parameters were: the number of principal components generated in the PCA method (from 1 to the minimum number of conditional attributes present in local tables $\min_i |A_i|$ was tested); the number of neurons in the hidden layer ($\{2, 2.5, \cdots, 9.5\}$ times the number of neurons in the input layer was tested). The experimental evaluation was performed on three data sets obtained from the UC Irvine Machine Learning Repository: Vehicle Silhouettes, Landsat Satellite, Dry Bean [15]. Each data available in the repository is stored in a single table. The training data sets are then dispersed into local tables. Different degrees of dispersion are considered in order to check whether the method can cope with significant data dispersion. The versions with 3, 5, and 7 local tables based on the original training set are considered where all local tables contained only a subset of the original set of conditional attributes. The evaluation of classification performance is conducted on the test set using the classification accuracy (*acc*), the

Figure 1. Model generation and prediction for test object stages



F-measure (F-m.)

$$\text{F-m.} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

and the balanced accuracy (*bacc*). All these metrics are calculated using functions implemented in Python in the sklearn library and using the weighted average option. All experiments were performed three times, and the results presented in Table 1 are the average value from these three performances. In the table the highest value in terms of accuracy obtained for different tested parameters values are given. As can be seen, the proposed model performed equally well with different degrees of dispersion. In addition, it can be noted that the F-measure is quite high compared to the accuracy. In papers [6, 7] the same data sets are studied using different approaches. For example, in paper [7] the PCA method was not used, and the MLP networks with the same structure were built using artificial objects

Table 1. Results of F-m., *bacc* and *acc* for the proposed approach.

No.	Vehicle			Satellite			Dry Bean		
tables	F-m.	*bacc*	*acc*	F-m.	*bacc*	*acc*	F-m.	*bacc*	*acc*
3	0.677	0.664	0.694	0.419	0.355	0.442	0.903	0.914	0.905
5	0.472	0.464	0.482	0.728	0.701	0.763	0.863	0.873	0.865
7	0.622	0.625	0.644	0.834	0.795	0.848	0.897	0.904	0.898

based on each local table separately. These networks were then aggregated into a single local network. Finally, the global network was post-trained. This approach is much more complex than the one presented in this paper, moreover, additional data are needed to post-train the global network (for details, see paper [7]). It can be concluded that the results presented in this paper are not significantly different from those presented in papers [6, 7]. In addition, the proposed approach is simpler, reduces data dimensionality and ensures data protection and privacy.

3. Conclusions

This paper presents a model for classification based on dispersed data. The system guarantees data protection and reduction of dimensionality by using the PCA for each local table separately. Based on these local models a single global table is created with additional information about the identifier of the table from which the data comes from. A global model is then built. MLP is used in the present study. The paper shows preliminary experimental results which indicates that the proposed model achieves good results. Of course, further experiments are needed to investigate more advanced network structures, as well as the use of different types of neural network models and feature extraction methods. However, preliminary experiments provide promising results.

References

[1] Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. 2016.

[2] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[3] Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

[4] Przybyła-Kasperek, M. and Marfo, K. F. Neural network used for the fusion of predictions obtained by the k-nearest neighbors algorithm based on independent data sources. *Entropy*, 23(12):1568, 2021.

[5] Przybyła-Kasperek, M. and Marfo, K. F. Studies on neural networks as a fusion method for dispersed data with noise. In *International Conference on Information Systems Development*, pages 169–186. Springer, 2022.

[6] Marfo, K. F. and Przybyła-Kasperek, M. Radial basis function neural network with a centers training stage for prediction based on dispersed image data. In *International Conference on Computational Science*, pages 89–103. Springer, 2023.

[7] Marfo, K. F. and Przybyła-Kasperek, M. Study on the use of artificially generated objects in the process of training mlp neural networks based on dispersed data. *Entropy*, 25(5):703, 2023.

[8] Ringnér, M. What is principal component analysis? *Nature biotechnology*, 26(3):303–304, 2008.

[9] Hu, P., Peng, D., Sang, Y., and Xiang, Y. Multi-view linear discriminant analysis network. *IEEE Transactions on Image Processing*, 28(11):5352–5365, 2019.

[10] Li, L., Fan, Y., Tse, M., and Lin, K.-Y. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[11] Huang, W., Li, T., Wang, D., Du, S., Zhang, J., and Huang, T. Fairness and accuracy in horizontal federated learning. *Information Sciences*, 589:170–185, 2022.

[12] Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.-Q., and Yang, Q. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[13] Zhu, H., Xu, J., Liu, S., and Jin, Y. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[14] Nguyen, H. T., Sehwag, V., Hosseinalipour, S., Brinton, C. G., Chiang, M., and Poor, H. V. Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218, 2020.

[15] Dua, D. and Graff, C. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *IEEE transactions on pattern analysis and machine intelligence*, 1(1):1–29, 2019.

Neural network pretraining and its influence on Continual Learning

Piotr Gręda[1], **Kamil Deja**[1,2]

[1]*Warsaw University of Technology*
piotr.greda.stud@pw.edu.pl
[2]*IDEAS NCBR*
kamil.deja@pw.edu.pl

Abstract. *Human brain can learn new tasks in a sequential fashion. This ability differentiates it from neural networks. Every new task causes a significant decrease in accuracy of previously learnt ones - this phenomenon is called "catastrophic forgetting". In this paper we investigate whether pretraining on a different dataset can boost a continual neural network's performance in new tasks. In particular, we investigate several pretraining approaches such as supervised learning, simulation of continual learning or self-supervised techniques. We compare evaluated techniques and draw the conclusions with practical suggestions on how to prepare the neural network for continual learning.*
Keywords: *Neural Networks, Continual Learning, Training, Pretraining*

1. Introduction

When adapting and acquiring knowledge related to additional or different categories e.g. classess neural networks tend to forget previously learned information. This happens because of the *catastrophic forgetting* [1]. Continual Learning (CL) methods aim to mitigate this problem. However, majority of them start the training from random weights initialization, what is impractical given recent state-of-the-art foundation models. In this work, we propose to study the effect of pretraining on the continually trained neural networks.

In particular, we compare several approaches for incorporating initial knowledge to the model. We start with the standard transfer learning methods between

DOI:10.17388/WUT.2024.0002.MiNI

different datasets, while we also investigate whether knowledge build in an unsupervised way provides better generalisation for future tasks. To that end we employ several self-supervised learning (SSL) [2] methods that use data structures and relationships, instead of labels, to differentiate between classes and objects.

In the experimental section, we check if pretraining can boost the performance of continually trained neural networks. To that end, we analyze whether different source datasets can give different results at the target dataset in the CL setup, and show which pretraining methods create knowledge structures that are easier to transfer between tasks. We run our experiments in two setups - *online* (when a model is trained using only one epoch) and *offline* (with more epochs).

We can summarise our observations as follows: 1) Continually trained neural networks perform better when pretrained in some manner, 2) Self-supervised fine-tuning enhances model performance in most cases, 3) Online learning presents a more challenging Continual Learning setup, thus emphasizing the importance of pretraining, 4) Pretrained models are less susceptible to the influence of catastrophic forgetting, resulting in greater consistency in terms of accuracy.

2. Related Work

Pretraining Neural network pretraining is a technique in machine learning where a model is initially trained on a source dataset before moving to the target one. We use pretraining to initialize model weights to make it perform better at another dataset. [3].

Continual Learning Continual Learning [4] assumes that whole dataset is not known at the beginning of training. Therefore, the goal of this approach is to learn task from additional data, without losing information on the previous ones.Early efforts to alleviate catastrophic forgetting [5] commonly involved incorporating a memory buffer [6], wherein a portion of the data from each task was stored. Subsequently, a combination of replayed and new data samples was utilized during the learning of each task. In the evolution of CL, many alternative approaches have been developed [7]. In order to overcome catastrophic forgetting [8], three main groups can be distinguished, each of them based on a different way of storing and using task-specific information: Some methods replay samples from previous tasks while learning a new one - i.e. Experience Replay [9] or Generative Replay [10]. Other approaches implement extra regularization terms in the loss function, aiding

in retaining previous knowledge - i.e. Learning without Forgetting [11] (LwF) or Elastic Weight Consolidation (EWC) [1]. Finally architectural methods expand the network to represent a new task or train only its specific part - i.e. CopyWeights with Re-init (CWR) [12]

Self-supervised learning In contrast to supervised learning, which is constrained by the presence of labeled data, self-supervised methods have the ability to glean knowledge from the extensive unlabeled datasets - visualised in Fig. 1. The model has no information about the label; it receives only data. Using that data, the model learns to differentiate one class from another.

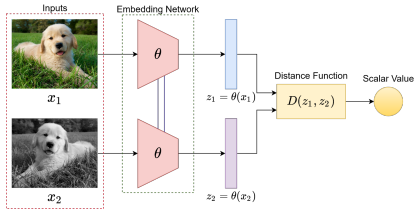


Figure 1: Joint embedding architecture, as an example of Self-Supervised Learning, image from [13].

Following [2], we define three main groups of self-supervised learning:

- The Deep Metric Learning Family (i.e. Nearest-Neighbor Contrastive Learning of visual Representations (NNCLR)[14]) is based on encouraging similarity between semantically transformed versions of an input - uses contrastive loss function [15], that implements similarity between versions of an input into a learning objective.
- The Self-Distillation Family (i.e. Bootstrap Your Own Latent (BYOL) [16]) relies on a mechanism of providing two different views (of the same image formed by image transformations including random resizing, cropping, color jittering, and brightness alterations [2]) to two encoders and mapping one to the other using predictor
- The Canonical Correlation Analysis Family's (i.e. BarlowTwins[17]) fundamental objective is to uncover the relationship between two variables by analyzing their cross-covariance matrices

3. Methodology

3.1. Problem statement

We have two datasets: source (A_S) and target (A_T). (A_S) is divided into classes (Y_S), meanwhile (A_T) is split into T tasks, each containing classes (Y_T). (A_S) and

(A_T) consist of disjoint classes: $y_1 \in A_P \leftrightarrow y_1 \notin A_T$. Each task consists of multiple batches. Each batch (γ) contains several images, and each image (x) is assigned to one exclusive class from the Y_T classes.

The goal of this project is to use a source dataset (A_S) in order to initialize weights ψ in a way that optimizes the performance of a model continually trained on a target dataset (A_T). The process of initializing weights is done by using different ways of training on A_S - we call this process pretraining because it takes place before we move onto the next stage - training. After initializing ψ, we use A_T to train a model in Continual Learning setup - we call it training on the target dataset. The final performance is measure over all tasks in the target dataset.

3.2. Weight initialization

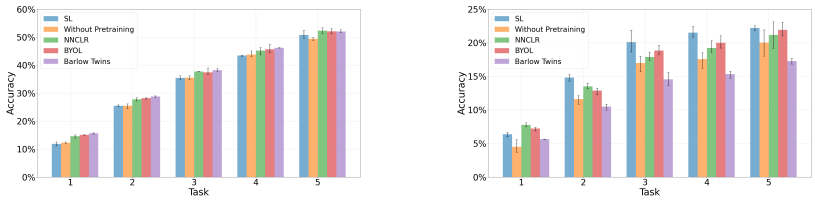
We investigate whether different weight initializations influence the accuracy at the end of the training. In particular we compare three main options:

- **Random initialization** – We initialize the model weights using kaiming uniform [18] strategy.
- **Supervised pretraining** – We train a simple classifier on the source dataset with a standard Cross-Entropy Loss.
- **Self-supervised pretraining** – We employ one of the three distinct methods, coming from a different SSL paradigm (deep metric learning, self-distillation or canonical correlation analysis) [2]. Each of these methods is used as a separate pretraining process, and subsequently, each pretrained model is utilized for training on the target dataset.

4. Results

In this section, we provide experimental comparison of the evaluated pretraining approaches. To that end, we use one of the two source datasets, CIFAR10 [19] and Mini-ImageNet100 [20]. We evaluate models pretrained on those datasets, with target ones, CIFAR100 [19] and CUB200 [21] respectively.

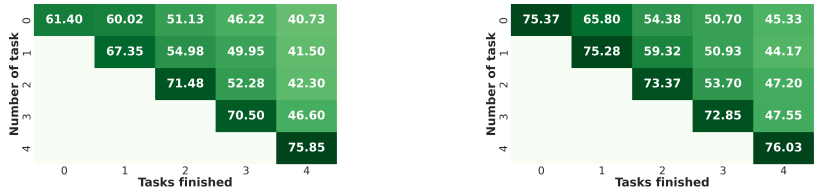
We start with a comparison of pretraining methods in the standard offline continual learning scenario. In Fig. 2a, we show that pretraining can be helpful in achieving higher accuracy at the end of the CL training. The model without any pretraining performs the worst. When comparing standard supervised pretraining



(a) Offline CIFAR10 → Split-CIFAR100

(b) Offline Mini ImageNet → Split-CUB200

Figure 2: Comparison between different pretraining methods in CIFAR10 → CIFAR100 and Mini ImageNet → CUB200 scenarios in the offline CL setup



(a) Accuracy on particular tasks on SplitCIFAR100 dataset without any pretraining - this plot shows *Catastrophic Forgetting* influence on neural networks.

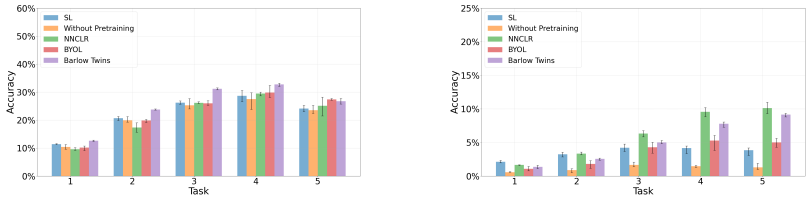
(b) Accuracy on particular tasks on SplitCIFAR100 dataset using Self-supervised pretraining (BYOL method) on SplitCIFAR10.

Figure 3: Impact of pretrainig on catastrophic forgetting

with SSL methods, we can see that initial knowledge build in semi-supervised way allows models to achieve higher accuracy on the target dataset, i.e., the SL pretrained model achieves 50.8% ± 1.63%, while NNCLR - the best performing method 52.1% ± 0.98%.

The effect of pre-training is more visible in online settings when the model has access to the new data samples only once (in Fig. 2b). The difference between the best performing, and baseline models in offline CIFAR100 setup, is 2.89%, while in online one it grows to 3.13%. This effect is even more visible with more complex datasets where the difference grows to 2.2% and 8.82%, respectively.

In continual learning scenarios, we can usually observe that the accuracy on previous tasks drops, as depicted in Fig. 3a. This is clearly illustrated in the last



(a) Online CIFAR10 → Split-CIFAR100 (b) Online Mini ImageNet → Split-CUB200

Figure 4: Comparison between different pretraining methods in CIFAR10 → CIFAR100 and Mini ImageNet → CUB200 scenarios in the online CL setup

column of the plot, where the accuracy for every task, after training on the entire dataset, is the lowest. When using SSL pretraining methods, we can see that the model achieves the highest accuracy on the current task (on a diagonal) and tends to lose less of the formerly gained knowledge when retrained on the next task. The first cell in Fig. 3b shows that even though the source dataset differed from the target one, the model learns relationships between classes and utilizes them when learning a new task.

Scaling to the bigger datasets In this section, we extend previous analysis to more complex datasets. Now, we use Mini Image-Net 100 to pretrain the model and CUB200 as the target dataset. In Fig. 4a, we present that pretraining still gives an advantage in the CL setup, allowing the model to reach higher accuracy.

In more complex setup, supervised pretraining allows the network to achieve the highest accuracy at the target dataset. It is the best method of all pretraining ones - contrary to what was observed in the previous scenario.

5. Conclusions

In this work, we investigate the influence of pretraining on Continual Learning. In our experiments, we show that pretraining the models is beneficial for CL, as it significatly improves the results and reduces the catastrophic forgetting. We experimentally validate that this effect is particularly visible when target dataset is challenging, and in the online setup. We also show that self-supervised learning as a pretraining method is worth considering.

References

[1] Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

[2] Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., and Goldblum, M. A cookbook of self-supervised learning, 2023.

[3] Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty, 2019.

[4] Ring, M. B. *Continual Learning in Reinforcement Environments*. Ph.D. thesis, USA, 1994. UMI Order No. GAX95-06083.

[5] McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi:https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

[6] Blakeman, S. and Mareschal, D. A complementary learning systems approach to temporal difference learning. *Neural Networks*, 122:218–230, 2020. ISSN 0893-6080. doi:https://doi.org/10.1016/j.neunet.2019.10.011. URL https://www.sciencedirect.com/science/article/pii/S0893608019303338.

[7] Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application, 2023.

[8] Lange, M. D., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. 2019. doi:10.1109/TPAMI.2021.3057446.

[9] Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. P., and Wayne, G. Experience replay for continual learning. *CoRR*, abs/1811.11682, 2018. URL http://arxiv.org/abs/1811.11682.

[10] Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[11] Li, Z. and Hoiem, D. Learning without forgetting. *CoRR*, abs/1606.09282, 2016. URL http://arxiv.org/abs/1606.09282.

[12] Lomonaco, V. and Maltoni, D. Core50: a new dataset and benchmark for continuous object recognition. In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 17–26. PMLR, 2017. URL https://proceedings.mlr.press/v78/lomonaco17a.html.

[13] The beginner's guide to self-supervised learning. https://www.v7labs.com/blog/self-supervised-learning-guide. Accessed: 2024-01-06.

[14] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, 2021.

[15] Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1. 2005. doi:10.1109/CVPR.2005.202.

[16] Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020. URL https://arxiv.org/abs/2006.07733.

[17] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. *CoRR*, abs/2103.03230, 2021. URL https://arxiv.org/abs/2103.03230.

[18] Pytorch documentation - torch.nn.init. https://pytorch.org/docs/stable/nn.init.html. Accessed: 2023-02-11.

[19] Krizhevsky, A. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

[20] Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., and Wierstra, D. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

[21] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. *The Caltech-UCSD Birds-200-2011 Dataset*. 2011.

Structure-aware normalization in unified models for abstract visual reasoning

Mikołaj Małkiński[0000−0002−5214−5928], **Jacek Mańdziuk**[0000−0003−0947−028X]

Warsaw University of Technology
Faculty of Mathematics and Information Science
Koszykowa 75, 00-662 Warsaw, Poland

mikolaj.malkinski.dokt@pw.edu.pl, mandziuk@mini.pw.edu.pl

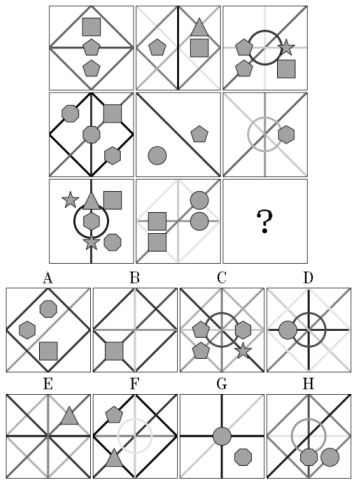
Abstract. *Recently, we proposed a new method to solve multiple Abstract Visual Reasoning (AVR) tasks that relies on a unified model architecture. In this work, based on temporal context normalization proposed by Webb et al., we extend the method with a structure-aware normalization layer that normalizes latent representations in a task-dependent context. Two normalization variants (parametric and non-parametric) are considered and shown to improve model performance across diverse AVR tasks.*
Keywords: *abstract visual reasoning, artificial intelligence, deep learning, multi-task learning*

1. Introduction

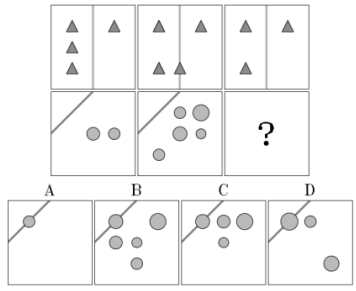
Recent research in artificial intelligence (AI) has studied the reasoning capabilities of deep neural networks in solving abstract visual problems [5, 6, 7, 8]. Tasks belonging to this domain, such as Raven's Progressive Matrices (RPMs) [9, 10], visual analogy problems (VAPs) [2], or odd-one-out (O3) tests [3], require the solver to identify abstract patterns in a 2D grid of images depicting geometric shapes. The goal of the solver is to fill-in the missing part of the matrix by selecting one of the provided choice panels (see Figure 1).

Recent progress in deep learning (DL) led to the development of task-specific methods that solve particular abstract visual reasoning (AVR) tasks with high effectiveness [6]. However, due to architectural constraints, the models are often

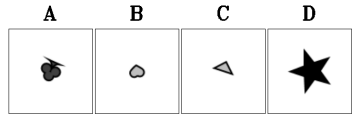
DOI:10.17388/WUT.2024.0002.MiNI



(a) PGM [1]



(b) VAP [2]



(c) O3 [3]

Figure 1: **AVR matrices.** In (a) and (b) the grid of context images (composed of 9 and 6 panels/images, resp.) has to be completed with the appropriate answer panel (respectively, A-H and A-D), while in (c) an odd panel has to be identified among the provided images. The matrices are governed by the following rules: (a) `OR` applied row-wise to shape position (lines serve as distractors); (b) `AND` abstracted from shape position to shape size; (c) all but one shapes have small size. The correct answers are C, A, and D, respectively. For additional examples, please refer to [4, Appendix C].

suited to solving specific tasks and can't be directly applied to solve other, even similar problems. In our recent paper, we propose to shift the focus of the AVR research towards universal methods capable of operating in multi-task settings [4]. Specifically, [4] introduces the unified model for solving Single-Choice Abstract visual Reasoning tasks (SCAR) and demonstrates its capability of tackling multiple AVR tasks with a unified model architecture.

A parallel stream of research investigates the importance of normalization in solving visual analogies. Specifically, temporal context normalization (TCN) [11] normalizes latent representations over a task-relevant temporal window. The method facilitates learning representations that reflect relative features of the analysed objects (e.g. their relative size). A subsequent work [12] applies TCN to a transformer-

based model and Scattering Compositional Learner [13]. In both cases, the use of TCN improves out-of-distribution generalization of the considered models.

In this work we seek to further improve the generalization capabilities of universal AVR models by incorporating normalization applied over task-specific window. To this end, we consider two variants of TCN and apply them to three variants of SCAR. The resultant models are evaluated on five AVR tasks: G-set [3], I-RAVEN [14, 15], PGM [1], VAP [2], and O3 tests [3]. Our experiments show that TCN is an effective method of improving the performance of universal AVR models.

2. Method

Let $(\{x_i\}_{i=1}^N, y)$ be an AVR matrix composed of N image panels, where $x_i \in [0,1]^{h \times w}$ is a greyscale image with height h and width w, and y is the index of the correct answer. AVR models are typically composed of a panel embedding module $\mathcal{P} : x \to h$ that produces a latent representation h_i of panel x_i and a reasoning module $\mathcal{R} : \{h\} \to y$ that predicts the answer y from the representations $\{h_i\}_{i=1}^N$. In this work, we implement $\mathcal{P}$ and $\mathcal{R}$ based on the SCAR architecture [4] and consider three variants of the model that differ in the initial layer of $\mathcal{R}$, referred to as the *fusion* layer. The explored choices include the Structure-Aware dynamic Layer (SAL) [4], Relation Network (RN) [16] and LSTM [17]. The layers are designed to operate on sets of objects of arbitrary size, which makes them suitable to tackle AVR tasks with diverse structures.

Following [4], panel embeddings $\{h_i\}$ are arranged into A groups $\{H_j\}_{j=1}^A$ each of size K, where A is determined by the number of choices (answer panels) in the considered matrix. We refer to the embeddings belonging to group H_j as $\{z_{jk}\}_{k=1}^K$. As an example, panel embeddings of RPMs from I-RAVEN are arranged into 8 groups by combining the context embeddings with each answer embedding. Differently from [4], we study the effect of normalization applied to panel embeddings in each group before processing them with $\mathcal{R}$. To this end, we employ TCN [11] defined as:

$$\boldsymbol{\mu}_k = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{z}_{jk} \qquad \boldsymbol{\sigma}_k = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{z}_{jk} - \boldsymbol{\mu}_k)^2 + \boldsymbol{\epsilon}} \tag{1}$$

$$TCN(\boldsymbol{z}_{jk}) = \gamma \frac{\boldsymbol{z}_{jk} - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} + \beta \tag{2}$$

Fusion	Normalization	Test accuracy (%)				
		G-set	PGM-S	I-RAVEN	VAP-S	O3
SAL	-	82.1	46.1	94.7	**92.9**	**86.7**
SAL	NP-TCN	71.9	47.1	95.0	92.8	51.7
SAL	P-TCN	**85.3**	**48.6**	**95.5**	92.5	56.3
RN	-	**36.5**	33.2	**56.1**	**90.5**	69.5
RN	NP-TCN	20.0	**35.2**	48.0	90.0	69.2
RN	P-TCN	20.0	12.5	42.5	90.2	**69.9**
LSTM	-	43.5	26.0	41.7	**90.7**	52.4
LSTM	NP-TCN	**53.2**	13.3	48.6	89.9	**63.4**
LSTM	P-TCN	31.2	**32.4**	**51.5**	89.5	**63.4**

Table 1: **Single-task learning.** Test accuracy of SCAR variants that differ in the panel embedding fusion layer and the normalization method. The models are evaluated in the single-task learning scheme. For each fusion layer, the best result for a given dataset is marked in bold, and the second best is underlined.

where β and γ are trainable parameters initialized as $\beta = 0$ and $\gamma = 1$, and $\epsilon = 10^{-8}$ is a constant. We consider two normalization variants: non-parametric (NP-TCN), in which the parameters are frozen during training, and parametric (P-TCN), in which the parameters are optimized jointly with the model.

3. Experiments

We evaluate the proposed models in a single-task learning scheme using five AVR datasets. This includes three RPM sets: G-set [3], PGM-S (a subset of the Neutral regime of PGM [1]), and I-RAVEN [15] (an improved version of RAVEN [14]). The datasets contain 660, 42K, and 42K training matrices, resp. In addition, we use VAP-S (a subset of the VAP dataset [2]) with 42K training instances, and O3 [3] with 600 training samples. We follow the dataset specification and experimental settings described in [4].

The results are presented in Table 1. Most notably, SAL with P-TCN achieves better results than the original version of SCAR on G-set (+3.2 p.p.), PGM-S (+2.5 p.p.), and I-RAVEN (+0.8 p.p.). This variant is also competitive on VAP-S (−0.4 p.p.), but worse on O3 (−20.4 p.p.). SAL with NP-TCN outcompetes the base variant on PGM-S (+1.0 p.p.) and I-RAVEN (+0.3 p.p.), but is worse in the remaining

cases. The results of the remaining model variants don't present a clear trend.

We conclude that structure-aware normalization realized by the parametric and non-parametric variants of TCN generally has potential to improve the performance of universal models for solving AVR tasks. However, in the conducted experiments the normalization-based models didn't outcompete the baseline in all cases. Further research is needed to develop a robust normalization method that would perform favourable across diverse AVR tasks.

4. Conclusions

In this paper, we applied two variants of TCN to the SCAR model with three different fusion layers. The experiments show that normalization of latent representations in a task-dependent context is a strong technique for boosting the performance of the universal AVR model across multiple tasks. Nevertheless, further development is required to improve robustness of the method. In future work we plan to study the effectiveness of normalization methods that take into account task-dependent context in a multi-task setting.

Acknowledgements

This research was carried out with the support of the Laboratory of Bioinformatics and Computational Genomics and the High Performance Computing Center of the Faculty of Mathematics and Information Science Warsaw University of Technology. Mikołaj Małkiński was funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

References

[1] Barrett, D., Hill, F., Santoro, A., Morcos, A., and Lillicrap, T. Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning*, pages 511–520. PMLR, 2018.

[2] Hill, F., Santoro, A., Barrett, D., Morcos, A., and Lillicrap, T. Learning to make analogies by contrasting abstract relational structure. In *International Conference on Learning Representations*. 2019.

[3] Mańdziuk, J. and Żychowski, A. DeepIQ: A human-inspired AI system for solving IQ test problems. In *2019 International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2019.

[4] Małkiński, M. and Mańdziuk, J. One self-configurable model to solve many abstract visual reasoning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38. 2024.

[5] Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., and Dowe, D. L. Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, 230:74–107, 2016.

[6] Małkiński, M. and Mańdziuk, J. Deep learning methods for abstract visual reasoning: A survey on raven's progressive matrices. *arXiv preprint arXiv:2201.12382*, 2022.

[7] Małkiński, M. and Mańdziuk, J. A review of emerging research directions in abstract visual reasoning. *Information Fusion*, 91:713–736, 2023.

[8] Mitchell, M. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

[9] Raven, J. C. Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive. *Master's thesis, University of London*, 1936.

[10] Raven, J. C. and Court, J. H. *Raven's progressive matrices and vocabulary scales*. Oxford pyschologists Press Oxford, England, 1998.

[11] Webb, T., Dulberg, Z., Frankland, S., Petrov, A., O'Reilly, R., and Cohen, J. Learning representations that support extrapolation. In *International Conference on Machine Learning*, pages 10136–10146. PMLR, 2020.

[12] Mondal, S. S., Webb, T. W., and Cohen, J. Learning to reason over visual objects. In *The Eleventh International Conference on Learning Representations*. 2023.

[13] Wu, Y., Dong, H., Grosse, R., and Ba, J. The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *arXiv preprint arXiv:2007.04212*, 2020.

[14] Zhang, C., Gao, F., Jia, B., Zhu, Y., and Zhu, S.-C. RAVEN: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5317–5327. 2019.

[15] Hu, S., Ma, Y., Liu, X., Wei, Y., and Bai, S. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1567–1574. 2021.

[16] Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30:4967–4976, 2017.

[17] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Simulating neurogenesis in artificial neural networks

Jakub Walczak[1][0000−0002−5632−9484],
Adam Wojciechowski[1][0000−0003−3786−7225]

[1]*Lodz University of Technology*
Institute of Information Technology
al. Politechniki 8, 93-590 Łódź, Polska
jakub.walczak@p.lodz.pl

Abstract. *Neuroscience suggests that neurogenesis is a process influencing our cognitive and memory capabilities by altering neural connections structure and information encoding way. It is, hence, an interesting process to be adapted in dynamic artificial neural networks. This paper roughly drafts the concept behind neurogenesis modelled using graph structures.*
Keywords: *artificial intelligence, neurogenesis, new neurons, neural networks*

1. Introduction

The neurological and psychological inspirations for artificial neural networks are apparent beyond any doubt. The notion of 'artificial neural networks' (referred in this proposal also as **ANN**) itself shows unquestionable motivation behind it — biological neural networks present in animals' (mostly, mammals) brains. Not only confirms the name the biological foundation behind ANN but also the structure. An artificial neuron's design resembles its original [1] being composed by: a body cell, dendrites, and an axon. The early methmetical model of an artificial neuron was introduced in [2] and improved in [3], whose authors — according to our knowledge — as the first ones used the name so well known currently: **preceptron**. Not much time after presenting the model of perceptron, the adaptive linear element model (ADALINE) was released [4]. The current model of an artificial neuron has not changed significantly since then.

DOI:10.17388/WUT.2024.0002.MiNI

Single artificial neurons form layers, and layers — networks that are currently the most recognised tool in artificial intelligence. Though rising in their capabilities and superior in some specialised task, ANN still fall behind general cognitive abilities of human's brain. I believe the effective model of forgetting mechanism could be a milestone towards increasing intellectual abilities of artificial neural networks. But let us move forward through the biological inspirations behind ANN.

The convolution mechanism is quite old and its application dates back to 1980 [5], when **Neurocognitor** compliant with Hubel & Weisel [6] visual cortex model was introduced. The convolution operator was then succesfully used in [7] for visual handwritten digits recognition but their current popularity convolution neural networks owe A. Krizhevsky et al. [8]. Intriguing fact is also similarity between convolution filter responses [8] and Gabor functions widely used in brain activity modelling [9], particularly in the context of visual cortex [10, 11, 12, 13, 14, 15].

Yet another ANN employed mainly for sequential data analysis is long-short term memory (LSTM) [16, 17], clearly inspired by the model of human memory introduced by [18]. The LSTM model mimics some of the memory mechanisms on the sample-basis, namely, an incoming sample's chunks remembering and forgetting. This cannot be directly generalised to the entire learning process of artificial neural networks.

Those were just samples of biological or psychological inspirations behind artificial neural networks. There are much more, among which attention mechanism [19], reinforcement learning [20, 21], and Hopfield models [22, 23] are worth mentioning.

2. Related works

2.1. Neurogenesis

While the examples explored above offer valuable insights into inspiration of artificial intelligence, they don't fully leverage the potential of biological mechanisms. One promising avenue for exploration is neurogenesis, the process of creating new neurons within the brain [24]. This process has been shown to significantly impact cognitive abilities, including memory formation and consolidation [25]. By dynamically altering network topology through the addition and removal of neural connections, neurogenesis not only introduces new elements but also influences how information is stored and processed.

2.2. Dynamic neural networks

Modelling neurogenesis in artificial neural networks aligns with the ongoing research on dynamic neural networks, which hold potential benefits such as improved efficiency and representational power [26] are just cases in point. Han et al. [26] presented a taxonomy of dynamic neural networks methods, categorising them into method relying on architectural changes and on parameter changes (like dynamic weights adjustment in inference mode [27] or weights prediction [28]). As neurogenesis is a complex process still actively investigated by neurologists and physiologists, it is undoubtedly related to structure changes [29]. Therefore, architecture-varying methods seem most relevant for capturing its essence at this stage.

Sufficient flexibility of a network cannot be provided by models like multilayer perceptron, convolution, or even attention mechanism. All of them suffer from limitations due to the rigid structure. In order to effectively model dynamic models, like neurogenesis, flexible tools are required. Graph neural networks emerge as a particularly suitable solution due to their inherent spatial, structural, and representational plasticity [30].

3. Proposed solution

Before diving into the details, let us defined a simulated neurogenesis network (SNG) as an unconstrained directed graph G (Eq. 1) allowing cycles and feedback loops, mirroring the structure of brain neural networks [31, 32].

$$G = (V, E) \tag{1}$$

A single vertex ($w_i \in V$) represents a single learnable artificial neuron capable of connecting to input, output, other hidden neurons, or even itself (Fig. 1).

Artificial neurogenesis can be defined as a continuous and iterative process driven by two interconnected operations:

1. generating a new vertex with initial location feature vector based on a set of existing nodes (Eq. 2)
$$V \leftarrow V \cup \{G(V)\} \tag{2}$$
2. continuous edges rearangement (Eq. 3)
$$E \leftarrow F(V, E) \tag{3}$$

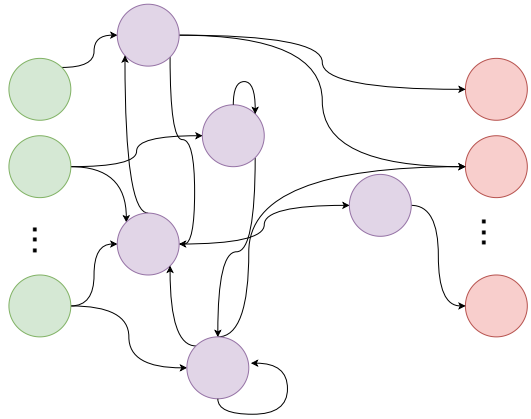


Figure 1. The concept structure of a SNG network snapshot. Learnable neurons (purple) can be connected to input neurons (green), output (red), other learnable neurons or to themselves.

The function G is the neuron generator function to be defined. A new neuron with an initial location feature vector emerges based on existing nodes in the network. Following neurological principles, G is triggered by exposure to new stimuli (e.g., a new sample in a dataset) [33, 34].

Edges are continuously adjusted (function F) by adding and pruning connections. This simulates the biological processes of neuron migration to their destination [35, 36] and integration with the network [37, 38]. In the computational model, the presence of an edge can be represented using fuzzy logic or a pseudometric function[1] where $d(w_i, w_j) = \infty$ means no arc from a vertex w_i to w_j, and $d(w_i, w_j) \neq \infty$ signifies the existence of an arc from w_i to w_j with weight equal to $d(w_i, w_j)$.

This example (Fig. 2) illustrates the core concept of our proposed method using a simplified network containing:

- one input neuron,
- one output neuron,
- one learnable neuron.

[1] With relaxing the symmetry axiom

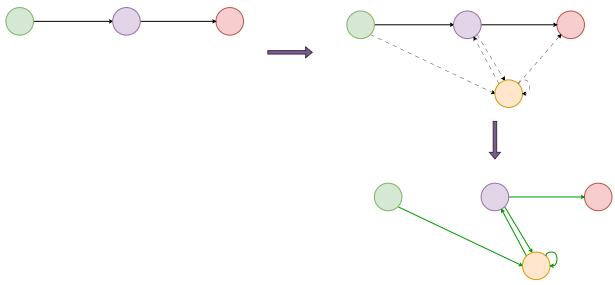


Figure 2. Top left: A balanced network. Top right: A new neuron introduces candidate connections that disturb the network. Bottom right: The network in the balanced state after integrating a new neuron.

In the initial state, the network is balanced, meaning it has successfully learned and produces the desired output.

Adding a new learnable neuron disrupts the network's equilibrium by establishing candidate connections. Initially, these connections have minimal impact, but they are enough to introduce instability.

To regain stability, the network undergoes an iterative process of edge rearrangement:

- some existing arcs are pruned,
- some of the candidate connections are established, integrating the new neuron into the network.

This process restores the network's equilibrium, allowing it to continue functioning effectively with the new neuron contributing to its capabilities (updated information encoding).

4. Conclusions

This paper introduces a novel approach to modelling artificial neurogenesis using graph networks. This method falls within the category of architecture-varying techniques for dynamic neural networks and holds potential for significant advancements. However, further research is required to validate its effectiveness and

demonstrate its encoding capabilities. Our upcoming work will specifically focus on addressing these aspects.

References

[1] Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S., Hudspeth, A. J., Mack, S., et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

[2] McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[3] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[4] Widrow, B., Hoff, M. E., et al. Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pages 96–104. New York, 1960.

[5] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[6] Hubel, D. H. and Wiesel, T. N. Brain mechanisms of vision. *Scientific American*, 241(3):150–163, 1979.

[7] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[9] Sinkkonen, J., Tiitinen, H., and Näätänen, R. Gabor filters: an informative way for analysing event-related brain activity. *Journal of Neuroscience Methods*, 56(1):99–104, 1995. ISSN 0165-0270. doi:https://doi.org/10.1016/0165-0270(94)00111-S. URL https://www.sciencedirect.com/science/article/pii/016502709400111S.

[10] Pollen, D. A., Nagler, M., Daugman, J., Kronauer, R., and Cavanagh, P. Use of gabor elementary functions to probe receptive field substructure of posterior inferotemporal neurons in the owl monkey. *Vision Research*, 24(3):233–241, 1984.

[11] Jones, J. P. and Palmer, L. A. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1233–1258, 1987.

[12] Hamada, T., Kato, K., and Okada, K. A model for development of gabor-receptive fields in simple cortical cells. *NeuroReport*, 7(3):745–748, 1996.

[13] Cho, Y. C. P., Bae, S., Jin, Y., Irick, K. M., and Narayanan, V. Exploring gabor filter implementations for visual cortex modeling on fpga. In *2011 21st International Conference on Field Programmable Logic and Applications*, pages 311–316. IEEE, 2011.

[14] Yu, Y., Hira, R., Stirman, J. N., Yu, W., Smith, I. T., and Smith, S. L. Mice use robust and common strategies to discriminate natural scenes. *Scientific reports*, 8(1):1379, 2018.

[15] Caballero-Puntiverio, M., Prichardt, S., Klem, L., Bundesen, C., Vangkilde, S., and Andreasen, J. Gabor patterns as stimuli in a rodent visual attention task. *Neuroscience Letters*, 728:134970, 2020.

[16] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[18] Atkinson, R. C. and Shiffrin, R. M. Human memory: A proposed system and its control processes. In *Psychology of learning and motivation*, volume 2, pages 89–195. Elsevier, 1968.

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Doya, K. Reinforcement learning: Computational theory and biological mechanisms. *HFSP journal*, 1(1):30, 2007.

[21] Subramanian, A., Chitlangia, S., and Baths, V. Reinforcement learning and its connections with neuroscience and psychology. *Neural Networks*, 145:271–287, 2022.

[22] Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[23] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[24] Begega, A., Alvarez-Suarez, P., Sampedro-Piquero, P., and Cuesta, M. Chapter 1-effects of physical activity on the cerebral networks a2-watson, ronald ross. physical activity and the aging brain (pp. 3–11), 2017.

[25] Apple, D. M., Fonseca, R. S., and Kokovay, E. The role of adult neurogenesis in psychiatric and cognitive disorders. *Brain Research*, 1655:270–276, 2017.

[26] Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2021.

[27] Yang, B., Bender, G., and Ngiam, J. Conditionally parameterized convolutions for efficient inference. In *Proceedings of the Computer Vision and Pattern Recognition*. 2019.

[28] Jia, X., De Brabandere, B., Tuytelaars, T., and Gool, L. V. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

[29] Azadian, M. M. and George, P. M. Neurogenesis. In *Reference Module in Neuroscience and Biobehavioral Psychology*. Elsevier, 2023. ISBN 978-0-12-809324-5. doi:https://doi.org/10.1016/B978-0-12-820480-1.00040-1. URL `https://www.sciencedirect.com/science/article/pii/B9780128204801000401`.

[30] Skarding, J., Gabrys, B., and Musial, K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

[31] Garrido, M. I., Kilner, J. M., Kiebel, S. J., and Friston, K. J. Evoked brain responses are generated by feedback loops. *Proceedings of the National Academy of Sciences*, 104(52):20961–20966, 2007.

[32] Rezaei, H., Aertsen, A., Kumar, A., and Valizadeh, A. Facilitating the propagation of spiking activity in feedforward networks by including feedback. *PLoS computational biology*, 16(8):e1008033, 2020.

[33] Nilsson, M., Perfilieva, E., Johansson, U., Orwar, O., and Eriksson, P. S. Enriched environment increases neurogenesis in the adult rat dentate gyrus and improves spatial memory. *Journal of neurobiology*, 39(4):569–578, 1999.

[34] Grońska-Pęski, M., Gonçalves, J. T., and Hébert, J. M. Enriched environment promotes adult hippocampal neurogenesis through fgfrs. *Journal of Neuroscience*, 41(13):2899–2910, 2021.

[35] Tsai, H.-H. and Miller, R. H. Glial cell migration directed by axon guidance cues. *Trends in neurosciences*, 25(4):173–175, 2002.

[36] Kaneko, N., Sawada, M., and Sawamoto, K. Mechanisms of neuronal migration in the adult brain. *Journal of neurochemistry*, 141(6):835–847, 2017.

[37] Nissant, A. and Pallotto, M. Integration and maturation of newborn neurons in the adult olfactory bulb–from synapses to function. *European Journal of Neuroscience*, 33(6):1069–1077, 2011.

[38] Rahimi-Balaei, M., Bergen, H., Kong, J., and Marzban, H. Neuronal migration during development of the cerebellum. *Frontiers in cellular neuroscience*, 12:484, 2018.

Chapter 5

Natural Language Processing, Automatic Speech Recognition, and Conversational AI

Track Chairs:

- prof. Maciej Piasecki - Wrocław University of Science and Technology
- prof. Piotr Pęzik - University of Lodz

Mining United Nations General Assembly Debates*

Mateusz Grzyb[1], **Mateusz Krzyziński**[1], **Bartłomiej Sobieski**[1], **Mikołaj Spytek**[1], **Bartosz Pieliński**[2[0000−0003−2664−2135]], **Daniel Dan**[3[0000−0002−7251−7899]], **Anna Wróblewska**[1[0000−0002−3407−7570]]

[1]*Warsaw University of Technology, Faculty of Mathematics and Information Science, Koszykowa 75, 00-662 Warsaw, Poland*
{mateusz.grzyb3, mateusz.krzyzinski, bartlomiej.sobieski, mikolaj.spytek}.stud@pw.edu.pl, anna.wroblewska1@pw.edu.pl
[2]*University of Warsaw, Faculty of Political Science and International Studies, Krakowskie Przedmieście 26/28, 00-927 Warsaw, b.pielinski@uw.edu.pl*
[3]*Modul University Vienna, School of Applied Data Science Am Khalenberg 1, 1019, Vienna, Austria daniel.dan@modul.ac.at*

Abstract. *This project explores the application of Natural Language Processing (NLP) techniques to analyse United Nations General Assembly (UNGA) speeches. Using NLP allows for the efficient processing and analysis of large volumes of textual data, enabling the extraction of semantic patterns, sentiment analysis, and topic modelling. Our goal is to deliver a comprehensive dataset and a tool (interface with descriptive statistics and automatically extracted topics) from which political scientists can derive insights into international relations and have the opportunity to have a nuanced understanding of global diplomatic discourse.*

Keywords: *natural language processing, information extraction, political language*

*This work was funded by the European Union under the Horizon Europe grant OMINO (grant number 101086321) and by the Polish Ministry of Education and Science within the framework of the program titled International Projects Co-Financed. This research was also carried out with the support of the Faculty of Mathematics and Information Science at Warsaw University of Technology and its High-Performance Computing Center.

1. Introduction

The United Nations (UN) is an international organization founded in 1945, comprising 193 member states. It was established after World War II with the intent to prevent future conflicts and foster global peace and security. The UN is a global forum where countries discuss and address critical issues ranging from international security, economic development, climate change, human rights, and humanitarian aid. It operates through various organs, including the General Assembly, the Security Council, and specialized agencies like UNESCO and WHO. The UN is pivotal in international cooperation and diplomacy, striving to maintain global stability and promote sustainable development.

The United Nations General Assembly (UNGA) serves as a global forum for member states to discuss and work together on international issues. It meets annually in regular sessions, with each member state having one vote. Essential functions include overseeing the UN's budget, appointing non-permanent members to the Security Council, and making recommendations in the form of General Assembly Resolutions. The United Nations General Assembly (UNGA) debate transcripts are official records of the speeches delivered during the UNGA's annual sessions. These transcripts date from 1946 and provide detailed accounts of the statements made by representatives from each member state, reflecting their positions on various global issues. The transcripts are essential for understanding the diplomatic stances of different countries, their interactions in the international arena, and the evolution of global policies and initiatives.

The huge volume of data source material makes manual analysis unfeasible. NLP methods came into play and allow extracting detailed information. We enriched the UNGA data with metadata and completed with the latest speeches. Our

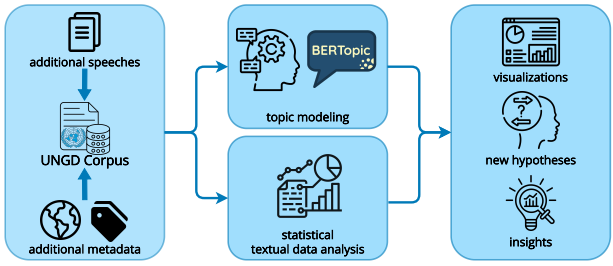


Figure 1. Diagram showcasing main steps of the project.

project spanned over three main building blocks – see Figure 1. The first one is data preparation, collecting the UNGA speeches between 1946-2023, completed with dates, names, the role of the speaker, and other additional features. The second is the exploration of the corpus and the calculation of speech statistics, together with preparing visualizations of the results. Third is the application of topic modelling techniques based on transformers.

2. Related work

The social and political science fields gained popularity lately due to the capability to process textual data [1] and the NLP application was a game changer [2, 3]. The value of more than 10,000 speeches from representatives of 193 countries represents the most extensive resource of its kind. [4] provided additional examples of how such corpus can be leveraged; topic modeling is one of the techniques. In social science, countless examples of NLP methods are applied, especially topic modeling harnessed to compare structured corpora to find semantic similarities or dissimilarities [5]. The most prominent technique in topic modelling is BERTopic [6], which is considered state-of-the-art. As there are no current frameworks based on AI that allow for comparison analysis, this work would fill in this gap.

3. Our approach

The dataset creation starts from an existing publicly available UNGD Corpus [7, 8]. The speeches are dated until 2022. The dataset was updated with the latest speeches extracted from the images through OCR software. Addressing data quality concerns, we correct errors and inconsistencies, particularly in the metadata. Moreover, our endeavor involves enriching the dataset's metadata by introducing additional variables and indices that provide a more comprehensive understanding of the characteristics of countries and their situation across different years. Furthermore, we integrate information from the United Nations geoscheme, providing division into regions and sub-regions.

In order to deliver automatic tools for topic modelling, we introduced BERTopic [6]. It uses a pretrained Large Language Model, which uses semantic similarity between the words in documents. Because it uses word embeddings to generate topics, the semantic similarity of documents plays an important role in

finding meaningful topics. In the main clustering process, BERTopic operates on uninterpretable word embeddings, and the human-understandable descriptions are extracted at the last step using the TF-IDF technique for each cluster. A metric is calculated for each word in each document, and the words with the highest scores are chosen as topic descriptions.

4. Experiments and results

Enhanced UNGD Corpus. The result of our work is the enhanced version of the United Nations General Debate Corpus, now inclusive of speeches delivered in 2023. Consequently, we added 120 speeches, constituting a subset of the 195 originally delivered. In total, the enhanced UNGD Corpus consists of 10,679 speeches. The enhancement also improved metadata integrity, achieved through an iterative feature engineering process, including various refinements, such as rectifying typographical errors, harmonizing nomenclature to standardize varied representations, and addressing inaccuracies in ISO codes. Further data engineering and analyses were facilitated by resolving complications related to countries that no longer exist. In total, 266 ISO codes, 1,862 country names, 3,877 names of speakers, and 3,199 of their positions were improved for enhanced accuracy. A significant augmentation in the new version of the UNGD Corpus is the inclusion of additional metadata, comprising 10 new covariates matched to all speeches whenever applicable. The data were drawn from diverse sources, including Gapminder, Our World in Data, and the United Nations, alongside its agencies and programs.[1]

BERTopic analyses. BERTopic was evaluated using different embedding methods. The following metrics have been used for evaluating the topic models: (1) **Topic coherence** [9, 10] metric utilizes various statistics drawn from the reference corpus and the split into topics, (2) **Topic diversity** [11] which is a much simpler metric calculated based solely on the extracted topics and evaluating how much variability there is among them. Table 1 and Table 2 show values of the topic coherence and diversity metrics, respectively.

By both metrics, the LDA method performs significantly worse than methods from the BERTopic family. Regarding the topic coherence metric, the DistilBERT

[1] `https://www.gapminder.org/data/`,`https://ourworldindata.org`, `http://data.un.org/datamartinfo.aspx`

is the best for any number of topics. Regarding topic diversity metrics, the DistilBERT method is the first or second best, depending on the number of topics.

Table 1. Topic coherence metric results.

# topics	Topic modeling method				
	LDA	BERTopic			
		all-MiniLM-L6-v2	all-MiniLM-L12-v2	all-mpnet-base-v2	DistilBERT
10	0.387	0.458	0.432	0.433	**0.463**
20	0.397	0.450	0.422	0.427	**0.499**
50	0.401	0.442	0.428	0.456	**0.493**

Table 2. Topic diversity metric results.

# topics	Topic modeling method				
	LDA	BERTopic			
		all-MiniLM-L6-v2	all-MiniLM-L12-v2	all-mpnet-base-v2	DistilBERT
10	0.250	**0.630**	0.500	0.560	0.570
20	0.250	0.445	0.430	0.450	**0.470**
50	0.258	0.358	0.342	**0.390**	0.372

Visualisations. After choosing the methods, considering topic coherence and topic diversity, the results were summed up in an interactive application developed using the Streamlit data app framework. This app allows political science scholars to navigate the corpus efficiently and conduct analyses based on their domain of interest.[2]

5. Conclusions

Our contributions can be summarized with a set of complementary approaches to enhancement and improvement of the UNGA debates corpus, development of an interactive application allowing easier extracting and summarising information, comprehensive tests on topic modelling using numerical metrics: BERTopic topic

[2]Our source code and links to binary dataset and our demo are available in our GitHub project

modeling pipeline applied together with 5 different embedding methods and compared with the LDA baseline using topic coherence and diversity metrics.

Notably, while the UN General Debates corpus was previously available, its utilization by non-technical users was hindered by limited ease of access. We believe that the refined user interface empowers researchers, policymakers, and other non-technical users, enabling them to explore and extract valuable insights from the data more efficiently and comprehensively than before.

References

[1] Hollibaugh, G. E. The use of text as data methods in public administration: A review and an application to agency priorities. *Journal of Public Administration Research and Theory*, 2019.

[2] Nay, J. NLP and machine learning for law and policy texts. 2018.

[3] Glavaš, G., Nanni, F., and Ponzetto, S. P. Computational analysis of political texts: bridging research efforts across communities. In *ACL: Tutorials*. 2019.

[4] Baturo, A., Dasandi, N., and Mikhaylov, S. J. Understanding state preferences with text as data: Introducing the un general debate corpus. *Research & Politics*, 2017.

[5] Valdez, D., Picket, A. C., Young, B. R., and Golden, S. On mining words: the utility of topic models in health education research and practice. *Health Promotion Practice*, 2021.

[6] Grootendorst, M. BERTopic: Neural Topic Modeling with a Class-based TF-IDF Procedure. *arXiv:2203.05794*, 2022.

[7] Jankin, S., Baturo, A., and Dasandi, N. *United Nations General Debate Corpus 1946-2022*, 2017.

[8] Dasandi, N., Jankin, S., and Baturo, A. Words to unite nations: The complete UN General Debate Corpus, 1946-present, 2023.

[9] Lau, J. H., Newman, D., and Baldwin, T. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*. 2014.

[10] Röder, M., Both, A., and Hinneburg, A. Exploring the space of topic coherence measures. In *WSDM*. 2015.

[11] Dieng, A. B., Ruiz, F. J., and Blei, D. M. Topic modeling in embedding spaces. *TACL*, 2020.

Deepfake tweets automatic detection*

**Adam Frej, Adrian Kamiński, Piotr Marciniak,
Szymon Szmajdziński, Soveatin Kuntur, Anna Wróblewska**

*Warsaw University of Technology,
Faculty of Mathematics and Information Science
{soveatin.kuntur.dokt, anna.wroblewska1}@pw.edu.pl*

Abstract. *This study addresses the critical challenge of detecting DeepFake tweets, leveraging advanced NLP techniques to differentiate between genuine and AI-generated texts. With the increasing prevalence of misinformation, our work utilizes the TweepFake dataset to train and evaluate a variety of machine learning models. The objective is to identify effective strategies for recognizing DeepFake content, thereby contributing to the integrity of digital communications.*
Keywords: *fake news detection, natural language processing, deepfake*

1. Introduction

The rise of DeepFake technology in the digital era presents both opportunities and challenges, significantly impacting misinformation through realistic fake content creation, especially in social media tweets [1, 2]. Our study leverages natural language processing (NLP) to develop a DeepFake tweet detection framework, aiming to bolster social media information reliability and pave the way for further research in ensuring digital authenticity.

*This work was funded by the European Union under the Horizon Europe grant OMINO (grant no 101086321) and by the Polish Ministry of Education and Science within the framework of the program titled International Projects Co-Financed. (However, the views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them.) This research was also carried out with the support of the Faculty of Mathematics and Information Science at Warsaw University of Technology and its High-Performance Computing Center.

Focusing on detecting deepfake content in tweets, this research employs the TweepFake dataset to evaluate various text representation and preprocessing methods. We explore effective embeddings and model-generated tweet patterns, utilizing machine learning, deep learning, and transformer technologies. The objective is to enhance the detection of GPT2-generated deepfakes, contributing to the development of robust detection algorithms.

2. Related Work

The TweepFake dataset's introduction has significantly advanced DeepFake detection research, enabling the assessment of different algorithms and highlighting transformer models like RoBERTa's effectiveness with an 89.6% accuracy [3]. This shift towards sophisticated language models, such as BERT, XLNet, and RoBERTa, over traditional bag-of-words, marks a significant progress in textual DeepFake identification by enhancing semantic comprehension [4][5]. Additionally, novel detection methods like Histogram-of-Likelihood Ranks and energy-based decoding have broadened the scope of detection techniques, though with varied success [6]. The extension of DeepFake detection research to other forms of digital text, including online reviews, further illustrates the widespread challenge of identifying synthetic content and the continuous need for innovative and versatile detection approaches [7].

3. Datasets

Our research utilized the TweepFake dataset [3] and GPT-2 generated texts [8], focusing on DeepFake detection. **TweepFake Dataset** [3] contains 25,572 tweets, split evenly between real tweets from 17 Twitter accounts and fake tweets from 23 bot accounts, offering a diverse set of fake tweet samples for analysis. **GPT-2 Generated Datasets** [8] comprises texts created by GPT-2 [9], with 500,000 training samples and 5,000 samples each for validation and testing, evenly split between human and machine-generated texts. These datasets support training and testing our models directly, facilitating a comprehensive evaluation of our detection techniques against varied DeepFake content.

4. Our Approach

Our study leveraged the TweepFake dataset and GPT-2 generated texts, adhering to an 80%-10%-10% split for training, validation, and testing, aiming for direct comparison with prior work [3]. We focused on:

1. Identifying effective text representations and preprocessing for TweepFake dataset deepfake detection.

2. Evaluating machine learning, deep learning, and transformer model efficacy in deepfake detection.
3. Assessing emoticons, mentions, misspells, and URLs' roles in distinguishing between human and automated tweets.
4. Enhancing the detection algorithm with advanced algorithm-generated deepfakes.

We explored various embeddings and bot-generated tweet patterns, utilizing NLTK [10] and SpacyTextBlob [11] for data insights. Preprocessing involved tokenization, stop word removal, and stemming or lemmatization. Our analysis spanned machine learning models (LightGBM [12], XGBoost [13], Random Forest [14], Logistic Regression [15], SVM [16]) with TF-IDF and BERT embeddings [17], deep learning networks (CNN, GRU, CNN+GRU), and transformers (xlm-roberta-base [18], distilbert-base-uncased [19], gpt2 [9]), focusing on accuracy, precision, recall, and F1 score to refine Twitter DeepFake detection.

5. Results

Table 1, catalogs the premier models by their balanced accuracy. Remarkably, the ROBERTA model, leveraging the unaltered TweepFake dataset (raw data in its original, unmodified state), emerged as superior in both balanced accuracy and F1 score. An evaluation across data processing techniques—specifically, unprocessed (raw) data, BERT embeddings (where text is converted into vectors capturing nuanced contextual relationships through the BERT algorithm), and lemmatization (simplifying words to their base form for uniformity)—demonstrated that raw data and BERT embeddings are particularly potent for deepfake detection.

Table 2 delves into the models' performance against different tweet sources, illustrating the nuanced challenge of identifying GPT2-engineered deepfakes. It becomes evident that RNN-generated tweets are more straightforward to detect, contrasting with the subtlety of GPT2-produced forgeries, which signifies their advanced deceptive quality.

Our investigations have yielded significant progress in the detection of deepfakes, especially those generated by GPT2, establishing new benchmarks in this complex arena. This advancement is crucial, considering the sophisticated nature of such fabrications. Figure 1 offers a visual exposition of the detection challenge posed by GPT2-generated deepfakes and the relative proficiency of transformer-based models in this domain.

6. Conclusion

Our study has shown that current models can accurately detect deepfakes made by GPT-2. Yet, we must recognize how quickly language models are advancing. GPT-2 was a big

Table 1. Top 10 DeepFake Detection Models Ranked by Accuracy

model	pre-processing	ba	f1	precision	recall
ROBERTA (TweepFake)	*raw*	***0.896***	***0.897***	*0.891*	*0.902*
XLM2	raw	0.8835	0.8821	**0.8934**	0.8711
SVC	bert	0.8757	0.8763	0.8729	0.8797
XLM1	raw	0.8713	0.8786	0.8328	**0.9297**
DISTIL_BERT0	raw	0.8698	0.8686	0.8773	0.8602
GPT2	raw	0.8671	0.8686	0.8593	0.8781
LGBM	bert	0.8561	0.8590	0.8429	0.8758
DISTIL_BERT1	raw	0.8554	0.8529	0.8681	0.8383
XGB	bert	0.8518	0.8546	0.8395	0.8703
CharCNN+GRU	lemmatized	0.8408	0.8518	0.7975	0.9141
LR	bert	0.8393	0.8416	0.8304	0.8531

Table 2. Model Accuracy Across Different Sources of DeepFakes

	TWEET CREATOR (CATEGORY)				
model name	ALL	GPT2	HUMAN	OTHERS	RNN
ROBERTA_FT (TweepFake)	***0.896***	*0.74*	*0.89*	*0.95*	***1.00***
XLM2_raw	0.8835	0.6953	**0.8959**	0.9153	0.9830
SVC_bert_embeddings	0.8757	0.6927	0.8717	0.9442	0.9782
XLM1_raw	0.8714	**0.8307**	0.8130	0.9607	0.9854
DisitlBERT0_raw	0.8698	0.6589	0.8795	0.9112	0.9879
GPT2_raw	0.8671	0.6693	0.8560	0.9587	0.9782
LGBM_bert_embeddings	0.8561	0.6745	0.8365	0.9483	0.9782
DistilBERT1_raw	0.8554	0.6849	0.8725	0.8471	0.9709
XGB_bert_embeddings	0.8518	0.6562	0.8333	0.9525	0.9733
CharCNN_GRU_lemmatized	0.8409	0.7760	0.7676	**0.9628**	0.9854
LR_bert_embeddings	0.8393	0.6380	0.8255	0.9236	0.9709

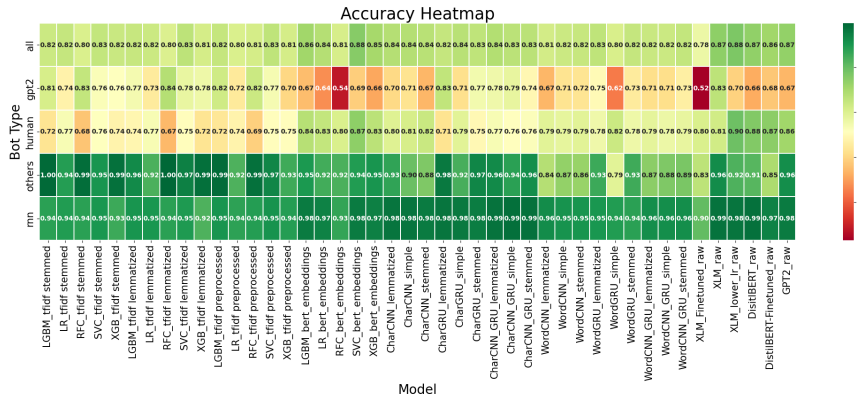


Figure 1. Accuracy for class type and every model

step forward, but newer, more complex models are constantly changing how we understand and generate language. This situation underscores the need for detection methods that keep pace with these advancements. As language models improve, telling real from fake content will become harder, highlighting the need for ongoing updates in detection methods. Future research should not only improve existing techniques but also prepare for newer model capabilities. This approach is vital for protecting the integrity of online communication as deepfake technology evolves.

References

[1] Veerasamy, N. and Pieterse, H. Rising above misinformation and deepfakes. In *International Conference on Cyber Warfare and Security*, volume 17, pages 340–348. 2022.

[2] Pawelec, M. Deepfakes and democracy (theory): how synthetic audio-visual media for disinformation and hate speech threaten core democratic functions. *Digital society*, 1(2):19, 2022.

[3] Fagni, T., Falchi, F., Gambini, M., Martella, A., and Tesconi, M. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):1–16, 2021. doi:10.1371/journal.pone.0251415.

[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019.

[5] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

[6] Gehrmann, S., Strobelt, H., and Rush, A. M. Gltr: Statistical detection and visualization of generated text, 2019.

[7] Adelani, D. I., Mai, H., Fang, F., Nguyen, H. H., Yamagishi, J., and Echizen, I. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. *CoRR*, abs/1907.09177, 2019.

[8] Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. Automatic detection of generated text is easiest when humans are fooled. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *ACL*, pages 1808–1822. Association for Computational Linguistics, Online, 2020. doi:10.18653/v1/2020.acl-main.164.

[9] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners, 2019.

[10] Bird, S., Klein, E., and Loper, E. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009. ISBN 0596516495.

[11] Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi:10.5281/zenodo.1212303.

[12] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[13] Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *ACM SIGKDD*, KDD '16, pages 785–794. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939785.

[14] Ho, T. K. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[15] Cox, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[16] Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[17] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[18] Conneau, A. e. a. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019.

[19] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

Generating Text Ads Based on Job Offers with GAN models

Drozda Paweł[1,2][0000−0003−3163−9408], **Maciej Osowski**[2][0000−0003−0277−3798]
Arkadiusz Nowacki[2][0009−0001−2290−871X]
Aleksandra Krasnodębska[2][0009−0004−1702−0865]

[1]*University of Warmia and Mazury in Olsztyn, Poland,*
pdrozda@matman.uwm.edu.pl
[2]*Emplocity S.A., Warsaw, Poland, https://emplocity.com*

Abstract. *This paper presents a new GAN solution for generating job advertisements using seq2seq technology and table-to-text method. The approach permits the definition of only a limited number of attributes, such as job location or required skills, yet it is able to generate the entirety of the advertisement content. In addition, We conducted the experiments on an extension of the seq2seq algorithm, incorporating an attention mechanism derived from Transformers. As a result, we have achieved a notable enhancement in the accuracy, thereby enabling the generation of semantically accurate advertisements. These advancements have been successfully implemented in the industry by Emplocity S.A.*
Keywords: *NLP, GAN, Ads Generation, Attention mechanism*

1. Introduction

One of the main challenges we face in our daily lives is looking for a job. This is often a very time-consuming and frustrating process. On the other hand, companies looking for employees devote a lot of energy and resources to reach the right job candidates. There is a lot of research to help automate the process. One direction supports matching candidates' resumes with job advertisements based on required skills and responsibilities [1]. Another solution has been proposed in [2] where Emplobot was introduced, automating part of the recruitment process. The research presented in this paper was aimed at developing methods for generating text-based job advertisements using GAN models, which were implemented as a

DOI:10.17388/WUT.2024.0002.MiNI

functionality of Emplobot. The prepared model, based on previous conversations with candidates, was able to prepare the appropriate content of recruitment job ads, where special emphasis was placed on syntactic and grammatical correctness. The use of GAN allowed the development of effective models that achieved a high level of precision.

2. GAN based job ads generator

GAN-type models [3] enable the generation of new data that did not exist in the original training data population, with properties similar to that data. This method is highly effective in the field of image generation [4]. In addition to such an application, attempts are also being made to use GAN models to generate text, although this is a more complex problem, as the generated sentences must have the appropriate grammatical structure and contain the correct conjugation of individual words. To address this issue, an architecture with the use of attention and seq2seq was developed. As a discriminator ULMFIT [5] was used to identify the generated content more accurate.

2.1. Table-to-text

As a result of the research, a model architecture was developed that can generate advertisements by transforming structured data in a table-to-text generation model, which made it possible to achieve high stability of results, much higher than, for example, in the text-to-text method (table 1). The right architecture for this type of network can also provide a kind of hierarchical solution and enforce certain predetermined patterns.

2.2. Attention and seq2seq

When using table-to-text, attention mechanism was needed. It gives the appropriate weight to the tokens, represented by slots (attributes), and to the position in the table as well [6]. It allows evaluation of the importance of given attributes in the content of the advertisement.

The advertisement content was created using the seq2seq architecture. This architecture was primarily designed for translation, conversational models, and text summarization problems [7]. In this particular case, it allowed for the generation of grammatically and inflectionally correct sentences, which was one of the main

Table 1. Table-to-text data example for GAN's ads generator

Company	Position	Benefits	Category	Location	Required skills
Atomy	bodyguard	attractive remuneration package	protection of persons and property	Gdańsk	availability
DataX	data scientist	remote work	research and development	Warsaw	python, machine learning

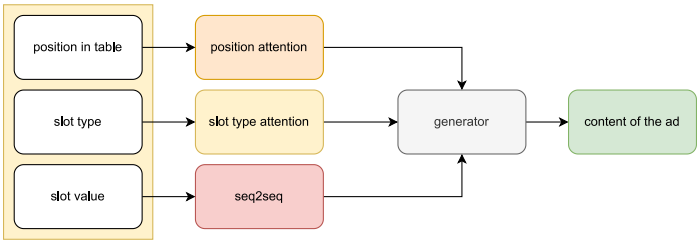


Figure 1. Model architecture using attentions and seq2seq modules

issues of the conducted research. The data can be of any form, numeric or text, although it must be in string form. All aforementioned elements are components of the designed architecture, presented in figure 1.

2.3. Discriminator

The Universal Language Model Fine-Tuning for Text Classification (ULMFiT) was used to discriminate between correct and incorrect content. This pre-trained language model learned to predict the next word in the sequence [5].

The trained model was then retrained on specific recruitment content to classify recruitment ads more accurately. Our approach of using a discriminator to classify well and poorly-constructed ad text has not been done before in the known literature on the subject. This is a more difficult task than the one typically used in the evaluation of text classifiers.

An additional advantage of ULMFIT as a discriminator is that learning with ULMFIT case transfer requires only about 2,000 described training samples for the entire architecture to achieve an F1-score close to manually created ads by humans. The model is also computationally efficient, which makes the discriminator model highly scalable and suitable for use in production environments.

2.4. Metrics

Our proposed metric for table-to-text reconstruction involves constructing a table-to-text for each entity in a generated paragraph and measuring precision, recall, and F-score by comparing it to the input table-to-text. This comparison is done from two perspectives: overall slot filling, where correct slots are identified when a slot type and its value are present in both the reconstructed and input table-to-text, and interdependent slot filling, where correct rows are identified when a row containing one or more slot types and their values are present in both the reconstructed and input table-to-text.

3. Results

Table 2 shows the metric values achieved by the model. The model's behavior was evaluated with and without attention mechanisms for type and slot. The results indicate that the use of attention mechanism significantly improves the recall metric, especially when comparing the seq2seq module alone to seq2seq with attention for type. Adding the attention mechanism known from Transformers significantly improves the model's ability to generate semantically correct texts, and the increase in the F1-score metric is significant. It is worth noting that the presented solution is novel, which makes it impossible to directly compare obtained findings with available experiments or studies. For this reason, an experiment was conducted to verify the value of the metrics for human-created ads. In this case, the value of the F1 score metric was 0.69.

Table 2. The results obtained for the designed architecture

Type	**Precision**	**Recall**	**F1-score**
sq2seq only	74.0	30.1	42.8
seq2seq with type attention	74.5	57.2	62.9
seq2seq with type and slot attentions	**74.9**	**61.4**	**67.5**

Training the model depends on the amount and range of training data. In the case of the generative model, we used a graphics processing unit (GPU) with 8GB memory to train the model. For instance, training with approximately 100k data described by 5 slots (attributes) with 50 iterations over the set takes about 3 hours. The loss graph is shown in Figure 2.

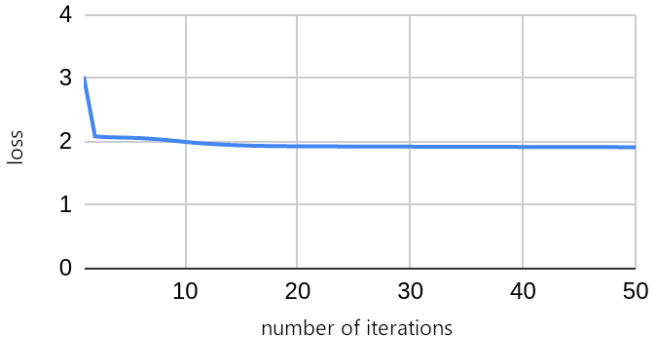


Figure 2. Loss function graph for the designed architecture

Based on the data available in the table (table 1), several variants of ads can be generated for the same job offer.

Using attributes such as company and position, we can generate following advertising: *Hi, I'm Emplobot and I'm looking for a job for people. I have a great offer for you: bodyguard. Click "Write message" and let's talk. No sending resumes or filling out forms.*
Don't wait - take off for success with Atomy!

In turn, using the attributes of position, benefits and location, the following content can be generated:

Get a job in Gdańsk! An attractive remuneration package waiting for you.
You don't need a resume, a short conversation with me on Messenger is enough.
Send me a message to apply!

3.1. LLM Comparison

The proposed solution was also compared and contrasted with Large Language Models [8], which are most commonly used in similar problems. For this purpose, a few-shot prompt was created, based on previously generated ads.

```
Generate ads based on keywords:

Keywords: Atomy, bodyguard
Ads: Hi, I'm Emplobot and I'm looking for a job for people. I
 ↪ have a great offer for you: bodyguard. Click "Write
 ↪ message" and let's talk. No sending resumes or filling out
 ↪ forms. Don't wait - take off for success with Atomy!

Keywords: bodyguard, attractive remuneration package, Gdańsk
Ads: Get a job in Gda nsk! An attractive remuneration package
 ↪ waiting for you. You don't need a resume, a short
 ↪ conversation with me on Messenger is enough. Send me a
 ↪ message to apply

Keywords: data scientist, remote work, DataX
Ads:
```

The same prompt was then applied to two large language models: GPT-3.5 and Mixtral 8x7B. The result for GPT 3.5 is as follows:

```
Unlock the world of remote opportunities as a Data Scientist
 ↪ with DataX! Enjoy the freedom of working from anywhere
 ↪ while diving deep into data. Skip the lengthy application
 ↪ process - shoot me a message to explore this exciting
 ↪ remote career. Let's reshape the future together!
```

and for Mistral 8x7B is as follows:

```
"Ready for a new challenge as a data scientist? Now offering
 ↪ remote work opportunities with competitive compensation at
 ↪ DataX. No resume needed - apply now through Messenger!"
"Join the DataX team as a data scientist from the comfort of
 ↪ your own home! Attractive remuneration package and
 ↪ flexible work arrangements available. No resume required -
 ↪ send me a message to learn more."
```

It can be seen that the developed model achieves similar results compared to widely used LLMs. In addition, it needs significantly fewer resources, which allows it to be used in a production environment.

4. Conclusions

This paper proposes using a GAN with ULMFIT as a discriminator to automatically generate job advertisements. The tests confirmed that the architecture can generate announcements with the specified attributes. The network can produce complete job advertisements that are suitable for social media or as a knowledge base for chatbots, even with residual information such as job location and required skills.

By utilizing ULMFIT as a discriminator, it is possible to rapidly increase the number of languages in which ads can be created, thanks to the multilingual version. This enables the creation of multilingual systems that generate job advertisements without the need for architecture redesign, thereby reaching a larger user base.

Large language models can generate attribute-based advertisements just as effectively without prior training. However, the resulting architecture requires less computation and therefore has a smaller carbon footprint than using LLMs.

Acknowledgments

This paper concerns the promotion of the results of the project No POIR.01.01.01-00-0912/18-00, “Development of an automated system using artificial intelligence for multi-channel recruitment advertising and Real-Time-Bidding operations using recurrent neural networks and Generative Adversarial Networks”, co-financed by the European Union.

References

[1] Hauff, C. and Gousios, G. Matching github developer profiles to job advertisements. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 362–366. IEEE, 2015.

[2] Drozda, P., Talun, A., and Bukowski, L. Emplobot-design of the system. In *Proceedings of the 28th International Workshop on Concurrency, Specification and Programming, Olsztyn, Poland*. 2019.

[3] Durgadevi, M. et al. Generative adversarial network (gan): a general review on different variants of gan and applications. In *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pages 1–8. IEEE, 2021.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[5] Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[8] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

LlamBERT: Large-scale low-cost data annotation in NLP

Bálint Csanády[1], **Lajos Muzsai**[1], **Péter Vedres**[1]
Zoltán Nádasdy[2,3], **András Lukács**[1]

[1]*ELTE Eötvös Loránd University, Institute of Mathematics, AI Research Group*
csbalint@protonmail.ch, muzsailajos@protonmail.com, vedrespeter0000@gmail.com, andras.lukacs@ttk.elte.hu
[2]*ELTE Eötvös Loránd University, Institute of Psychology*
[3]*The University of Texas at Austin, Department of Psychology*
zoltan@utexas.edu

Abstract. *Large Language Models (LLMs), such as GPT-4 and Llama 2, show remarkable proficiency in a wide range of natural language processing (NLP) tasks. Despite their effectiveness, the high costs associated with their use pose a challenge. We present LlamBERT, a hybrid approach that leverages LLMs to annotate a small subset of large, unlabeled databases and uses the results for fine-tuning transformer encoders like BERT and RoBERTa. This strategy is evaluated on two diverse datasets: the IMDb review dataset and the UMLS Meta-Thesaurus. Our results indicate that the LlamBERT approach slightly compromises on accuracy while offering much greater cost-effectiveness.*
Keywords: *NLP, data annotation, LLM, Llama, BERT, ontology, artificial intelligence*

1. Introduction

In the contemporary technological landscape, when confronted with the task of annotating a large corpus of natural language data using a natural language prompt, LLMs such as the proprietary GPT-4 [1] and the open-source Llama 2 [2] present themselves as compelling solutions. Indeed, minimal prompt-tuning enables them to be highly proficient in handling a wide variety of NLP tasks [3]. However,

DOI:10.17388/WUT.2024.0002.MiNI

running such LLMs on millions of prompts demands large and expensive computational resources. There have been optimization efforts aimed at achieving superior performance with reduced resource requirements [4, 5]. Numerous studies have investigated the efficiency and resource requirements of LLMs versus smaller transformer encoders and humans [6, 7, 8, 9, 10, 11]. Recent advancements in data augmentation with LLMs [12] underscore our approach, which relies on data labeling. Going beyond the exclusive use of LLMs for a task, we combine LLMs with substantially smaller yet capable NLP models. A study closest to our approach is [13], where GPT-NeoX was used to surrogate human annotation for solving named entity recognition.

Through two case studies, our research aims to assess the advantages and limitations of the approach we call LlamBERT, a hybrid methodology utilizing both LLMs and smaller-scale transformer encoders. The first case study examines the partially annotated IMDb review dataset [14] as a comparative baseline, while the second selects biomedical concepts from the UMLS Meta-Thesaurus [15] to demonstrate potential applications. Leveraging LLM's language modeling capabilities, while utilizing relatively modest resources, enhances their accessibility and enables new business opportunities. We believe that such resource-efficient solutions can foster sustainable development and environmental stewardship.

2. Approach

Given a large corpus of unlabeled natural language data, the suggested LlamBERT approach takes the following steps: (i) Annotate a reasonably sized, randomly selected subset of the corpus utilizing Llama 2 and a natural language prompt reflecting the labeling criteria; (ii) Parse the Llama 2 responses into the desired categories; (iii) Discard any data that fails to classify into any of the specified categories; (iv) Employ the resulting labels to perform supervised fine-tuning on a BERT classifier; (v) Apply the fine-tuned BERT classifier to annotate the original unlabeled corpus.

We explored two binary classification tasks, engineering the prompt to limit the LLM responses to one of the two binary choices. As anticipated, our efforts to craft such a prompt were considerably more effective when utilizing the 'chat' variants of Llama 2 [16]. We investigated two versions: `Llama-2-7b-chat` running on a single A100 80GB GPU, and `Llama-2-70b-chat` requiring four such GPUs. We also tested the performance of `gpt-4-0613` using the OpenAI API.

3. The IMDb dataset

The Stanford Large Movie Review Dataset (IMDb) [14] is a binary sentiment dataset commonly referenced in academic literature. It comprises 25,000 labeled movie reviews for training purposes, 25,000 labeled reviews designated for testing, and an additional 50,000 unlabeled reviews that can be employed for supplementary self-supervised training. This dataset serves as a fundamental baseline in NLP for classification problems, which allows us to evaluate our method against a well-established standard [17, 18, 19].

3.1. Experimental results

All of the results in this section were measured on the entire IMDb sentiment test data. In Table 1, we compare the performance of Llama 2 and GPT-4 in different few-shot settings. Due to limited access to the OpenAI API, we only measured the 0-shot performance of GPT-4. The results indicate that the number of few-shot examples has a significant impact on `Llama-2-7b-chat`. This model exhibited a bias toward classifying the reviews as positive, but few-shot examples of negative sentiment effectively mitigated this. Likely due to reaching the context-length limit, 3-shot prompts did not outperform 2-shot prompts on `Llama-2-7b-chat`, achieving an accuracy of 87.27%. The inference times shown in Table 1 depend on various factors, including the implementation and available hardware resources; they reflect the specific setup we used at the time of writing.

Table 1: Comparison LLM test performances on the IMDb data.

	Accuracy %			**Inference time**		
LLM	**0-shot**	**1-shot**	**2-shot**	**0-shot**	**1-shot**	**2-shot**
Llama-2-7b-chat	75.28	89.77	93.93	3h 54m	4h 16m	8h 14m
Llama-2-70b-chat	95.39	95.33	95.42	28h 11m	39h 6m	76h 2m
gpt-4-0613	96.40	N/A	N/A	49h 11m	N/A	N/A

In Table 2, we compare various pre-trained BERT models that were fine-tuned for five epochs on different training data with a batch size of 16. First, we established a baseline by using the original gold-standard training data. For the LlamBERT results, training data labeling was conducted by the `Llama-2-70b-chat` model from 0-shot prompts. The LlamBERT results were not far behind the baseline measurements, underscoring the practicality and effectiveness of the frame-

work. Incorporating the extra 50,000 unlabeled data in LlamBERT resulted in a slight improvement in accuracy. We also evaluated a combined strategy where we first fine-tuned with the extra data labeled by `Llama-2-70b-chat`, then with the gold training data. The large version of RoBERTa performed the best on all 4 training scenarios, reaching a state-of-the-art accuracy of 96.68%. Inference on the test data with `roberta-large` took 9m 18s, after fine-tuning for 2h 33m. Thus, we can estimate that labeling the entirety of IMDb's 7.816 million movie reviews [20] would take about 48h 28m with `roberta-large`. In contrast, the same task would require approximately 367 days on our setup using `Llama-2-70b-chat`, while demanding significantly more computing power.

Table 2: Comparison BERT test accuracies on the IMDb data.

BERT model	**Baseline train**	**LlamBERT train**	**LlamBERT train&extra**	**Combined extra+train**
distilbert-base [21]	91.23	90.77	92.12	**92.53**
bert-base	92.35	91.58	92.76	**93.47**
bert-large	94.29	93.31	94.07	**95.03**
roberta-base	94.74	93.53	94.28	**95.23**
roberta-large	96.54	94.83	94.98	**96.68**

3.2. Error analysis

To assess the relationship between training data quantity and the accuracy of the ensuing BERT model, we fine-tuned `roberta-large` across different-sized subsets of the gold training data as well as data labeled by `Llama-2-70b-chat`. As the left side of Fig. 1 indicates, the performance improvement attributed to the increasing amount of training data tends to plateau more rapidly in the case of LlamBERT. Based on these results, we concluded that labeling 10,000 entries represents a reasonable balance between accuracy and efficiency for the LlamBERT experiments in the next section. We were also interested in assessing the impact of deliberately mislabeling various-sized random subsets of the gold labels. The discrepancy between the gold-standard training labels and those generated by Llama 2 stands at 4.61%; this prompted our curiosity regarding how this 4.61% error rate compares to mislabeling a randomly chosen subset of the gold training data. As shown on the right side of Fig. 1, `roberta-large` demonstrates substantial resilience to random mislabeling. Furthermore, data mislabeled

by `Llama-2-70b-chat` results in a more pronounced decrease in performance compared to that of a random sample.

Table 3: Comparison of human annotation to model outputs on wrong test answers.

RoBERTa	**LlamBERT train**			**Combined extra+train**		
sentiment	positive	negative	mixed	positive	negative	mixed
positive	31	16	13	25	17	13
negative	17	14	9	15	14	16

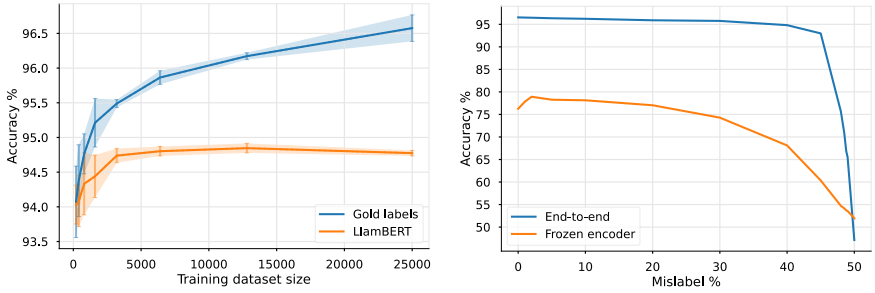


Figure 1: Accuracy (%) comparison of RoBERTa classifiers on the IMDb test data. On the left: The effects of training data size. On the right: The effects of intentionally mislabeling a random part of the gold training data.

We also conducted a manual error analysis on two of the models fine-tuned from `roberta-large`. For the model fine-tuned with the combined strategy, we randomly selected 100 reviews from the test data, where the model outputs differed from the gold labels. We sampled an additional 27 mislabeled reviews of the model fine-tuned with the LlamBERT strategy to get a sample size of 100 on the errors of this model too. We collected human annotations for the sentiment of the selected reviews independently from the gold labels. In the case of human annotation, we added a third category of *mixed/neutral*. Reviews not discussing the movie or indicating that 'the film is so bad it is good' were typically classified in this third category. Table 3 compares the human annotations to the model outputs. The results indicate a comparable ratio of positive to negative labels between the human annotations and the model outputs, suggesting that the model outputs are more aligned with human sentiment than the original labels. Overall human performance on this hard subset of the test data was worse than random labeling.

4. The UMLS dataset

The United Medical Language System (UMLS) [15], developed by the United States National Library of Medicine, is a comprehensive and unified collection of nearly 200 biomedical vocabularies. It has played a crucial role in fields such as natural language processing, ontology development, and information retrieval for over 30 years [22]. The UMLS Metathesaurus consolidates various lexical variations of terms into single concepts, outlining their interrelationships. However, its breadth, with over 3 million concepts, complicates the selection of specific subsets for research due to its vague semantic labels. Faced with the need to identify a distinct subset of the Metathesaurus for subsequent research, we aimed to classify anatomical entities within it, based on their relevance to the human nervous system. Previous research on creating a neurological examination ontology involved extracting terms from case studies and manually mapping them to UMLS concepts, a task that can be extremely labor-intensive [23]. Our approach streamlines this process by efficiently leveraging the vast amount of knowledge condensed into LLMs and mitigates the need for expert annotation.

By selecting relevant semantic types spanning multiple biological scales, but excluding genes, we were able to reduce the number of concepts to approximately 150,000 anatomical structures, resulting in a still substantially large dataset. Among these anatomical structures, we sought to find concepts related to the human nervous system, excluding purely vascular or musculoskeletal structures, and indirectly related entities such as the outer ear and eye lens. Using distinct random samples, we annotated 1,000 concepts for testing and an additional 1,000 for hand-labeled fine-tuning. We opted for a 1-shot prompt, on which `Llama-2-7b-chat` achieved an accuracy of 87.5%, while `Llama-2-70b-chat` reached 96.5%, and `gpt-4-0613` scored 94.6%. For fine-tuning BERT models, we labeled a distinct set of 10,000 concepts with `Llama-2-70b-chat`.

4.1. Experimental results

As shown in Table 4, fine-tuning general BERT models on the baseline hand-labeled dataset already yielded commendable results, however, our LlamBERT approach further improved these outcomes. Moreover, the combined strategy marginally surpassed Llama 2's initial performance. Within the biomedical domain, specific BERT models such as `BiomedBERT-large` [24] were already accessible and predictably outperformed both `bert-large` and `roberta-large` across

all training scenarios. Yet, the combined approach using `roberta-large` demonstrated comparable performance, suggesting that our methodology could serve as an alternative to training domain-specific models.

Table 4: Accuracy comparison of different training data for the UMLS classification; 95th percentile confidence interval measured on 5 different random seeds.

Model	Baseline	LlamBERT	Combined
bert-large	94.84 (±0.25)	95.70 (±0.21)	96.14 (±0.42)
roberta-large	95.00 (±0.18)	96.02 (±0.12)	96.64 (±0.14)
BiomedBERT-large	96.72 (±0.17)	96.66 (±0.13)	96.92 (±0.10)

5. Conclusions

Through two case studies showcasing the LlamBERT technique, we demonstrated the feasibility of efficiently labeling large quantities of natural language data with state-of-the-art LLMs. Combining the LlamBERT technique with fine-tuning on gold-standard data yielded the best results in both cases, achieving state-of-the-art accuracy on the IMDb benchmark. Our code is available on GitHub[1].

To further increase the quality of data initially provided by the LLM annotation, we aim to incorporate PEFT [25] techniques such as LoRA [26], prefix tuning [27], and P-tuning [28] in the future.

Acknowledgments

The authors thank the support of the National Research, Development and Innovation Office within the framework of the Thematic Excellence Program 2021 – National Research Sub programme: “Artificial intelligence, large networks, data security: mathematical foundation and applications" and the Artificial Intelligence National Laboratory Program (MILAB). We appreciate the support provided by OpenAI under the Researcher Access Program. We thank Máté Márk Horváth and Virág Bálint for their assistance in labeling the UMLS test dataset.

[1] `https://github.com/aielte-research/LlamBERT`

References

[1] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[3] Mao, R., Chen, G., Zhang, X., Guerin, F., and Cambria, E. GPTEval: A survey on assessments of ChatGPT and GPT-4. *arXiv preprint arXiv:2308.12488*, 2023.

[4] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[5] Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[6] Su, H., Kasai, J., Wu, C. H., Shi, W., Wang, T., Xin, J., Zhang, R., Ostendorf, M., Zettlemoyer, L., Smith, N. A., et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.

[7] Yu, H., Yang, Z., Pelrine, K., Godbout, J. F., and Rabbany, R. Open, closed, or small language models for text classification? *arXiv preprint arXiv:2308.10092*, 2023.

[8] Gilardi, F., Alizadeh, M., and Kubli, M. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi:10.1073/pnas.2305016120. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2305016120`.

[9] Savelka, J. and Ashley, K. D. The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6, 2023.

[10] Alizadeh, M., Kubli, M., Samei, Z., Dehghani, S., Bermeo, J. D., Korobeynikova, M., and Gilardi, F. Open-source large language models outper-

form crowd workers and approach ChatGPT in text-annotation tasks. *arXiv preprint arXiv:2307.02179*, 2023.

[11] Sprenkamp, K., Jones, D. G., and Zavolokina, L. Large language models for propaganda detection. *arXiv preprint arXiv:2310.06422*, 2023.

[12] Ding, B., Qin, C., Zhao, R., Luo, T., Li, X., Chen, G., Xia, W., Hu, J., Luu, A. T., and Joty, S. Data augmentation using llms: Data perspectives, learning paradigms and challenges. *arXiv preprint arXiv:2403.02990*, 2024.

[13] Frei, J. and Kramer, F. Annotated dataset creation through large language models for non-english medical NLP. *Journal of Biomedical Informatics*, 145:104478, 2023.

[14] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics, Portland, Oregon, USA, 2011. URL `http://www.aclweb.org/anthology/P11-1015`.

[15] Bodenreider, O. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[16] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[17] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[18] Heinsen, F. A. An algorithm for routing vectors in sequences. *arXiv preprint arXiv:2211.11754*, 2022.

[19] Wang, S., Fang, H., Khabsa, M., Mao, H., and Ma, H. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*, 2021.

[20] IMDb statistics. https://www.imdb.com/pressroom/stats/. Dec 2023.

[21] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[22] Jing, X. The unified medical language system at 30 years and how it is used and published: systematic review and content analysis. *JMIR Medical Informatics*, 9(8):e20675, 2021.

[23] Hier, D. B. and Brint, S. U. A neuro-ontology for the neurological examination. *BMC Medical Informatics and Decision Making*, 20:1–9, 2020.

[24] Chakraborty, S., Bisong, E., Bhatt, S., Wagner, T., Elliott, R., and Mosconi, F. BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679. 2020.

[25] Pu, G., Jain, A., Yin, J., and Kaplan, R. Empirical analysis of the strengths and weaknesses of PEFT techniques for LLMs. *arXiv preprint arXiv:2304.14999*, 2023.

[26] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRa: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[27] Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[28] Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68. 2022.

T^2R^2: Train, test, record, repeat: incremental framework for NLP model training

Nina Żukowska, Antoni Solarski, Oskar Szudzik, Radosław Winiecki, Mikołaj Morzy[0000−0002−2905−9538]

Poznan University of Technology
Faculty of Computing and Telecommunications
Piotrowo 2, 60-965 Poznań, Poland
Mikolaj.Morzy@put.poznan.pl

Abstract. *In this paper, we introduce an iterative training loop framework for natural language processing models, facilitating the seamless integration of various data-building tools for training, testing, and validation sets. Our approach empowers researchers and practitioners with flexible model training capabilities, enhanced by connectors to an assortment of modern NLP resources. Additionally, the framework integrates MLOps features such as automatic versioning, ensuring reproducibility, and streamlining the model development lifecycle. By enabling continuous refinement and evaluation, our solution paves the way for more robust and accurate NLP models that can be adapted to dynamic real-world datasets.*
Keywords: *deep learning, iterative training, MLOps, NLP*

1. Introduction

The field of Natural Language Processing (NLP) has recently experienced significant growth, transforming various industries. Central to this evolution is the Transformer architecture [1], which has revolutionized NLP through Large Language Models (LLMs). As NLP continues to advance, the importance of the model training process, the quality of training datasets, and structured workflows becomes evident. Although attention is often focused on model architecture and training, high-quality and representative datasets are crucial for robust machine

learning models. In this paper, we introduce the T^2R^2 framework[1], which addresses the challenges of maintaining flexibility, reproducibility, and traceability within NLP workflows. T^2R^2 offers a novel approach to model development that allows dynamic exploration of machine learning solutions.

2. NLP Iterative Training Loop

The iterative training loop in NLP is a cyclical process designed to enhance the performance and learning capabilities of a model through repetition and refinement. It includes selecting and modifying training and validation datasets, choosing suitable models and parameters based on the task, implementing learning techniques, and determining appropriate performance metrics. This methodology is akin to a greedy local search within the solution space, with the aim of generalization and robustness against unseen data through diversity in data selection. Datasets are carefully curated to lay the foundation for model training, which includes diverse and complex data gradually introduced to the model.

During training, the model is evaluated after each epoch using validation data, ensuring constant adaptation and improvement, with the training phase often requiring substantial time and monitoring tools. Post-training, the model is tested on a separate dataset to assess its generalization capability, based on predetermined metrics. These insights lead to potential adjustments in the training approach, and the loop is re-initiated for further refinement. The iterative nature of this process ensures a dynamic and continuous evolution in the development of machine learning models, allowing incremental improvements until the desired performance level is achieved.

The iterative training loop in NLP experiments is plagued by several challenges that hinder its practical application and prevent the broad exploration of various modeling techniques [2]. One of these challenges is the need for "glue code" due to the disparity between different APIs, which introduces added complexity and difficulty when it comes to testing. In addition, selecting the most appropriate model and learning technique within strict time constraints is a constant challenge. The integrity of the training datasets also poses a challenge, as it has a substantial impact on model performance. Furthermore, there are issues around recording and reproducibility, which encompass managing version control and documentation, as well as addressing problems associated with dead code paths. Finally, the com-

[1]Documentation and code examples are available here: `https://github.com/radswn/t2r2`

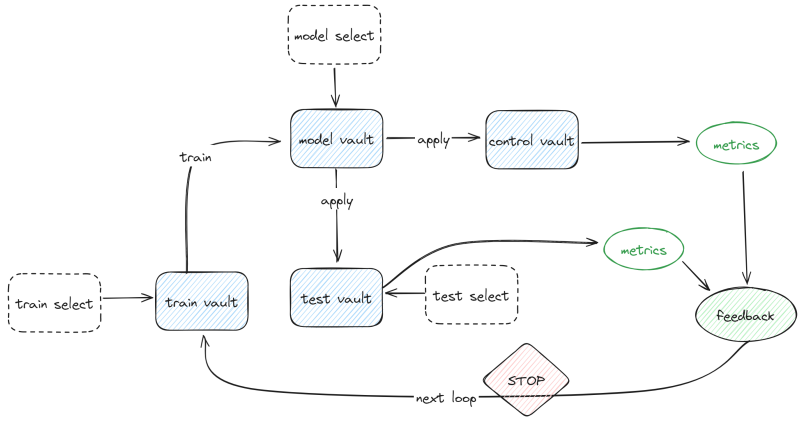


Figure 1. Conceptual overview of the system and iterative training loop.

plexity of configuration is exacerbated by hyperparameter tuning, which is made more difficult by the dispersal of configuration information, contributing to what is known as configuration debt [3].

Integrating ethical considerations into NLP development is complex, often being sidelined until after NLP models are trained. The central issue is that AI behavior is shaped by the training data it learns from. Biases within these data can lead to discriminatory outcomes, as the model can learn and perpetuate existing social and historical disparities. To counteract this, methods to promote fairness [4] in machine learning models include detailed descriptions of datasets and tools for evaluating model performance [5, 6, 7, 8]. Our proposal is to create *control dataset*, a set of critical examples that must be correctly identified by the model to verify ethical integrity. Errors in processing this control dataset would serve as a warning of possible fairness concerns. This approach advocates for the use of *platinum labels*, analogous to established gold labels, offering researchers a simple yet effective tool to ensure ethical model behavior during the training process.

3. System Architecture

The envisioned system utilizes a configuration-driven approach, with all training loop settings contained in a single YAML configuration file. This well-structured

file specifies model parameters, datasets, training, and evaluation criteria, ensuring the reproducibility of the training loop. The constituent elements of the system include the following:

- *Vaults*: For reproducible machine learning experiments, datasets must be tracked and versioned within a repository, or "vault," that includes training, test, and control datasets, as well as a separate vault for storing models, all managed using a version control system. This ensures that users can revert to specific iterations to access relevant datasets and model versions [2].
- *Selects*: The Model Select feature offers a pretrained model and the necessary tools, such as a tokenizer, for language processing. Users select pretrained models from Hugging Face or upload their own models. The various Selects streamline training by refining the CSV datasets and the training process. Train Select manages parameters such as the number of epochs and the batch size. Train Select also enables data selection techniques, for instance, random sampling or oversampling. The Test Select offers metrics analysis for particular test data subsets.
- *Feedback*: The user specifies which metrics to compute in the test and control datasets within a configuration file. These metrics, including those that track the training process, are calculated to generate feedback. The user reviews this feedback to assess the process and inform subsequent iterations, updating the configuration for further experimentation.

4. Tools and techniques

The T^2R^2 framework consists of several tools that implement the architectural components described in Section 3. Here, we briefly describe the tools used within the T^2R^2 framework.

- **Hugging Face (HF)**: Providing a variety of models from prominent organizations such as OpenAI, Google, and Facebook, the HF `transformers` library aids in tasks ranging from text classification to language generation and offers tools for efficient tokenization. Through the Hugging Face Hub, users can access a repository of models and datasets with detailed model cards. The `transformers` library is designed for convenience in the implementation of NLP workflows, including the fine-tuning or training models from scratch.

- **Data Version Control (DVC)**: DVC is a library that brings version control features from software development into machine learning and data science projects. It helps to track and manage workflows, particularly by focusing on data files, models, and code to facilitate collaboration, reproducibility, and project integrity. By interfacing with systems such as Git, DVC handles large data files efficiently and supports the iterative training of machine learning models by creating snapshots of the entire experiment configuration for each iteration.

- **MLFlow**: MLFlow is a widely used open-source MLOps platform designed to streamline machine learning development, including experiment tracking, reproducibility, and deployment. MLFlow provides an essential service for monitoring iterative training loops, enabling developers to log, monitor, and compare various model training sessions, hyperparameters, and metrics. MLflow supports the entire ML lifecycle, offering tools for traceability, consistent model access, and deployment, all while being agnostic to ML frameworks, which facilitates integration into existing pipelines.

- **Weak supervision**: Weak supervision is an approach in machine learning that uses less precise, but more scalable methods to create training data, such as labeling functions provided by experts, as opposed to manually labeling individual instances. These functions apply rules for tasks such as sentiment analysis or employ smaller models to generate a larger, albeit noisier, dataset. Weakly supervised learning includes a less discussed technique called *slice-based learning* [9], which focuses on monitoring the critical data subsets and improving performance in these subsets.

- **Curriculum Learning (CL)**: CL is a machine learning approach modeled after the way humans learn, where training data are presented in an organized way of increasing complexity [10, 11]. This strategy starts with simpler instances to build a strong foundational understanding before introducing more complex ones, potentially enabling faster convergence and better performance. As a type of transfer learning, curriculum learning allows a model to first master easier tasks before tackling more difficult target tasks. Various techniques such as self-paced learning and transfer teacher methods have been developed to optimize this learning process and have shown particular benefits for training language models.

- **Dataset Cartography (DC)**: DC is a method for assessing and improving the quality of large datasets used in language processing, with an emphasis on shifting the focus from data set size to data set quality [12]. The approach uses *data maps*, which plot instances based on two measures from a model's training dynamics: confidence (average predicted probability of the correct label) and variability (standard deviation of that probability). Through these dynamics, training instances can be classified into easy-to-learn, hard-to-learn, and ambiguous, based on their high or low confidence and variability.

- **Large Language Models (LLMs)**: LLMs have displayed impressive natural language processing prowess, coupled with emergent abilities that surpass those of smaller models [13]. Often used in a variety of text generation tasks, LLMs effectively aid in enhancing datasets through oversampling, undersampling, or custom manipulations to meet specific user requirements.

5. Conclusions and Future Work

In this paper, we have presented T^2R^2, a comprehensive library designed to streamline incremental training in NLP models through a configuration file that allows users to specify models, hyperparameters, data selectors and metrics. T^2R^2 stands out for its user-friendly approach to conducting NLP experiments, offering a selection of models from Hugging Face and integrated tools such as dataset cartography and curriculum learning for model optimization. The library can be used in text-based tasks, but it is designed for expansion into other ML domains and more diverse NLP tasks. One promising feature is the potential of LLMs for question-answering and summarization. The synergy between dataset cartography and curriculum learning presents another avenue to improve the efficiency of training. Future improvements to T^2R^2 will offer more configurability options such as loss functions and optimizers. Technical advances, such as enhanced code test coverage, will ensure robustness against new code changes. In its current state, the library provides a solid foundation for reproducible research with room for growth and innovation that could make it a mainstay in the future of machine learning experimentation.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Kreuzberger, D., Kühl, N., and Hirschl, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 2023.

[3] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015.

[4] Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., and Vertesi, J. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68. 2019.

[5] Bender, E. M. and Friedman, B. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018.

[6] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229. 2019.

[7] Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[8] Pushkarna, M., Zaldivar, A., and Kjartansson, O. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1776–1826. 2022.

[9] Chen, V., Wu, S., Ratner, A. J., Weng, J., and Ré, C. Slice-based learning: A programming model for residual learning in critical data slices. *Advances in neural information processing systems*, 32, 2019.

[10] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. 2009.

[11] Wang, X., Chen, Y., and Zhu, W. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2021.

[12] Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*, 2020.

[13] Larsen, B. and Narayan, J. Generative ai, a game-changer that society and industry need to be ready for. In *The World Economic Forum*. 2023.

Chapter 6

Computer Vision

Track Chairs:

- prof. Leszek Chmielewski - Warsaw University of Life Sciences
- prof. Krzysztof Gajowniczek - Warsaw University of Life Sciences
- prof. Bogdan Kwolek - AGH University of Krakow

Elevating point-based object detection in UAVs: A deep learning method with altitude fusion

Michał Wiliński[0009−0004−4818−8417], **Bartosz Ptak**[0000−0003−1601−6560], **Marek Kraft**[0000−0001−6483−2357]

Poznan University of Technology
Faculty of Control, Robotics and Electrical Engineering
Piotrowo 3A, 60-965 Poznań, Poland
marek.kraft@put.poznan.pl

Abstract. *Recent advancements in computer vision and deep learning have revolutionised remote sensing. An important challenge lies in detecting small objects like individuals in crowds from low-altitude drone-captured images, which are problematic due to scale variations. While existing research addresses the scale challenges, not enough focus has been put on the exploitation of altitude data from UAV sensors. This paper proposes three deep learning-based methods to fuse altitude and Ground Sampling Distance (GSD) information, which enhance input images with additional features and showcase significant improvements in point-oriented object detection on our custom dataset.*

Keywords: *remote sensing, deep learning, GSD and altitude fusion, data fusion*

1. Introduction

In recent years, convolutional neural networks (CNNs) and deep learning have significantly surpassed traditional computer vision algorithms in terms of performance. This progress is also notable in remote sensing tasks involving images captured by Unmanned Aerial Vehicles (UAVs). Researchers are increasingly focusing on the development of these methods due to their applicability across various domains, such as road safety [1] and Smart Cities initiatives [2]. One key area of focus involves the detection of tiny objects, such as individual persons

DOI:10.17388/WUT.2024.0002.MiNI

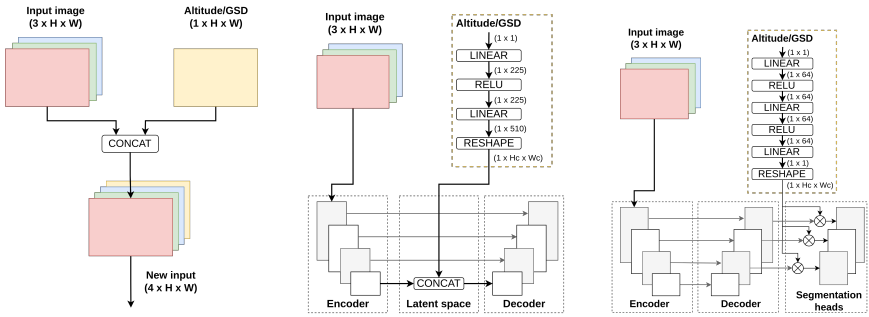
within a crowd, using low-altitude aerial imagery captured by drones. Unlike typical closed-circuit television (CCTV) applications, algorithms for drone imagery analysis must effectively address challenges like scale variations, perspective distortions, and movement-induced changes. Previous research has proposed various methods to address the performance drop due to scale variability of objects in remote-sensing images. However, none of these methods have leveraged direct altitude information from UAV sensors to enhance the task of detecting tiny objects, which may come as a surprise, since this information is usually directly and easily available.

In this paper, we propose and compare three methods dedicated to fusing altitude information, thereby enhancing input images with additional features. For each method, we evaluate the effectiveness of both raw altitude information and Ground Sampling Distance (GSD) to determine the most efficient fusion approach. Our experiments on the private dataset demonstrate a significant improvement compared to the baseline model in the task of detecting point-oriented objects.

2. Related Work

The integration of numerical information with image data, often referred to as data fusion, has been a subject of interest in various fields. One notable article that explores this intersection is [3]. This study explores different methodologies for integrating electronic health records (EHR), consisting of numerical and categorical features, with medical imaging. The authors identify and discuss three primary fusion strategies: early fusion, performed before inputting data into the model; joint fusion, where intermediate representations from different modalities are combined; and late fusion, which involves aggregating predictions from modality-specific models. Another work [4] proposes attentional feature fusion as a scheme for integrating features in modern network architectures, addressing issues related to inconsistent semantics and scales.

In the field of remote sensing, numerous studies have delved into data fusion. In [5], the authors analysed the data fusion capabilities for multisource data, concentrating solely on fusing images from different sensors. Another instance of data fusion is demonstrated in [6], where a two-step approach is employed. Initially, a deep learning model is trained for GSD estimation using images. Subsequently, the feature vector of the model is combined in the latent space to enhance the object detection task. Unexpectedly, none of these methods explored the direct impact of fusing altitude and GSD information available directly from the sensors.



(a) Method A - fuse raw information as an additional input channel.

(b) Method B - fuse as learnable features in latent space.

(c) Method C - fuse as learnable features at decoder stages level.

Figure 1: Overview of altitude information fusion methods. ALT - altitude information; GSD - Ground Sampling Distance.

3. Methods

Following state-of-the-art methods for point-oriented object detection, we employ the UNet [7] architecture with the EfficientNet-b2 [8] backbone, including weights pretrained on the ImageNet dataset. This model processes input images to produce an output mask where non-zero values represent estimated detection coordinates. Furthermore, to enhance the training process, we implement Deep Supervision, a widely recognised technique in remote sensing renowned for its ability to capture fine-grained details. The model takes images and resizes them to the resolution of 960×544 and generates masks in the same resolution, which are interpolated to the original resolution in the final step. The AdamW optimiser with the changing learning rate was applied in the training process, wrapping in the CosineAnnealingLR scheduler with an initial learning rate of $3e^{-4}$. We also perform augmentations (Flip, Rotate, Noise, Colour shift, Random Gamma) to improve the model's generalisation capabilities using varied images.

This architecture and training procedure are used as a baseline for our study. The methods outlined below follow the same training procedure, differing only in architecture and fusion methodology. Fig. 1 provides a general overview of the proposed methods.

Method A. In this approach, an additional channel is concatenated to the input

image, wherein the values represent normalised altitude or GSD. This approach stands out as the most efficient and straightforward means of integrating numerical data with input images. A visual representation of this process is depicted in Fig. 1a.

Method B. An alternative approach involves manipulation within the model's latent space (Fig. 1b). In this method, we suggest the incorporation of a learnable altitude or GSD embedding with a single latent channel size. Subsequently, this embedding undergoes resizing to align with the dimensions of the corresponding feature maps at that level and is appended as an additional feature map. This adaptation facilitates the development of a trainable altitude/GSD representation.

Method C. Finally, we focused on the intersection of the decoder and segmentation heads. As shown in Fig. 1c, we augmented the output of each decoder stage by performing element-wise product between output feature maps and learnable embedding of normalised GSD or altitude values. Subsequently, these augmented feature maps serve as inputs to the segmentation heads.

4. Evaluation

4.1. Dataset

We assess the method using an in-house dataset comprising 10000 images captured by the DJI Mini 2 drone at various altitudes ranging from 26.0 to 101.0 meters, with an average altitude of 60.3 meters. The dataset provides a list of coordinates delineating the human head for each image, along with corresponding altitude information indicating the capture height and is divided into train (6893), validation (1391), and test (1716) splits. Thanks to known altitude and camera parameters, we can calculate the GSD. We illustrate sample images featuring labelled heads in Fig. 2.

4.2. Results

In assessing the compared methods, we employ point-oriented detection metrics including Precision, Recall, and F1-Score, which are computed for test subsets. It is crucial to highlight that accurate detection is defined by an Euclidean distance between points that is equal to or less than 5 pixels in original size.

The experiment results are placed in Tab. 1. In general, the Ground Sample Distance (GSD) consistently outperformed altitude in terms of overall results. De-

Figure 2: Two example images with drawn head labels (red circle) on extremes of altitudes (left: 26.0m; right: 101.0m).

spite altitude exhibiting a marginally higher precision of 0.731, the GSD demonstrated better recall with a score of 0.737. Method A and B emerged as the frontrunners in this study, showcasing the best overall F1-score of 0.712. Furthermore, Method B's performance in achieving the highest F1-score for altitude reached a value of 0.711. All methods demonstrated a notable increase in performance compared to the baseline, indicating that the fusion of altitude or GSD data contributes to enhancing the model's performance.

Table 1: Methods comparison considering raw altitude and GSD fusion.

	Altitude			**GSD**		
Method	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Baseline	0.697	0.725	0.701	0.697	0.725	0.701
Method A	0.721	0.701	0.705	0.698	0.737	**0.712**
Method B	0.731	0.703	0.711	0.700	0.734	**0.712**
Method C	0.685	0.737	0.705	0.712	0.718	0.710

5. Conclusions

In this paper, we present that integrating altitude data with input imagery enhances performance metrics for detecting point-oriented objects in low-altitude aerial images. Moreover, our experiments reveal that combining Ground Sampling Distance (GSD) achieves better results compared to using altitude alone. We also show that the fusion implemented at the encoder is more effective rather than at the decoder level.

Data availability. We declare the dataset is not published within the article.

References

[1] Outay, F., Mengash, H. A., and Adnan, M. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transportation research part A: policy and practice*, 141:116–129, 2020.

[2] Kharchenko, V., Kliushnikov, I., Rucinski, A., Fesenko, H., and Illiashenko, O. UAV fleet as a dependable service for smart cities: Model-based assessment and application. *Smart Cities*, 5(3):1151–1178, 2022.

[3] Huang, S.-C., Pareek, A., Seyyedi, S., Banerjee, I., and Lungren, M. P. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *Digital Medicine*, 3, 2020.

[4] Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y., and Barnard, K. Attentional feature fusion. In *2021 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 3559–3568. 2021.

[5] Ghamisi, P., Rasti, B., Yokoya, N., Wang, Q., Hofle, B., Bruzzone, L., Bovolo, F., Chi, M., Anders, K., Gloaguen, R., et al. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39, 2019.

[6] Yang, Y., Wang, C., Cai, Z., Song, P., Huang, G., Cheng, M., and Zang, Y. GSDDet: Ground sample distance guided object detection for remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[7] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Intern. conf. on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[8] Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conf. on machine learning*, pages 6105–6114. PMLR, 2019.

Detection of Shoplifting Activities in video using SlowFast Architecture

Lyudmyla Kirichenko[1,2][0000−0002−2780−7993],
Bohdan Sydorenko[2][0000−0002−5963−5911],
Sergiy Yakovlev[3,4][0000−0003−1707−843X],
Oksana Pichugina[4][0000−0002−7099−8967]

[1] *Lodz University of Technology, Institute of Mathematics*
Politechniki 8, 90-924 Lodz, Poland
lyudmyla.kirichenko@p.lodz.pl
[2] *Kharkiv National University of Radio Electronics*
Nauki 14, 61166 Kharkiv, Ukraine
bogdan.sydorenko1@nure.ua
[3] *Lodz University of Technology, Institute of Information Technology*
Politechniki 8, 90-924 Lodz, Poland
sergiy.yakovlev@p.lodz.pl
[4] *National Aerospace University "Kharkiv Aviation Institute"*
Vadima Manka 17, 61070 Kharkiv, Ukraine
o.pichugina@khai.edu

Abstract. *The recognition of human activities in video surveillance holds numerous applications across diverse domains. This paper introduces an innovative approach to identifying shoplifting incidents within camera-recorded video data. The proposed methodology involves the application of human tracking to extract video segments featuring human activities. Subsequently, the video clips undergo categorization based on their respective activities. The detection of individuals is accomplished through the utilization of YOLO-NAS, while DeepSORT is employed for tracking purposes. Activity classification is carried out using PySlowFast. This integrated framework enhances the precision of shoplifting detection and demonstrates the efficacy of leveraging state-of-the-art computer vision techniques for comprehensive video analysis.*

Keywords: *human activity recognition, surveillance, shoplifting, yolo-nas, deepsort, pyslowfast*

DOI:10.17388/WUT.2024.0002.MiNI

1. Introduction

In modern times, video surveillance plays a crucial role in ensuring the security of public spaces. With the increase in the number of video cameras in cities and stores, effectively monitoring all recordings has become essential. Closed Circuit Television (CCTV) systems generate vast amounts of video data, necessitating automated processing. Thanks to video analytics technologies, recordings can be automatically analyzed to identify suspicious activities and moments requiring security intervention.

Various machine learning algorithms are employed for this purpose and trained on large datasets of criminal activities. Such models can recognize typical behavioural patterns of offenders and respond accordingly. This enables the surveillance system to detect thefts and other unlawful actions automatically.

Many studies emphasize the effectiveness of using Convolutional Neural Networks (CNNs) in this field, particularly in combination with other models such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM). Combinations of such models allow for consideration of both spatial and temporal characteristics of videos, making them effective for detecting various actions and behaviour scenarios [1-4]. However, in recent years, new tools and algorithms have emerged that efficiently address the task of identifying shoplifting cases [5-7].

This research aims to establish an end-to-end pipeline for identifying shoplifting incidents using video data from security cameras. To accomplish this goal, several state-of-the-art neural networks and algorithms, namely YOLO-NAS [8], DeepSort [9], and SlowFast [6], are being integrated. These algorithms are known for their effectiveness in various domains, but in this study, they are specifically adapted and optimized for the task of detecting shoplifting.

2. Methods and materials

The developed pipeline incorporates human detection using the state-of-the-art pre-trained YOLO-NAS (You Only Look Once - Neural Architecture Search) object detection neural network, observation through the DeepSORT (Deep Association Metric SORT) tracking algorithm, and extraction of video segments featuring a specific individual. These segments are then classified using a fine-tuned PySlowFast model into two classes: 0 - non-shoplifting and 1 - shoplifting.

For training, validation, and testing, the authors utilized a proprietary dataset

created by actors in conditions approximating real-world retail environments, captured from 10 different perspectives. The prepared training split of the classification dataset comprises 14 084 video clips with durations ranging from 3 to 5 seconds. Artificial augmentation techniques, including grid distortion, horizontal and vertical rotations, flipping, color jitter, and blurring, were employed to augment the training dataset artificially, expanding it by a factor of 11. This process generated approximately 154,924 video clips.

The YOLO-NAS and DeepSort combination proves to be a potent solution for tracking people in videos. YOLO-NAS is an object detection algorithm pretrained on the COCO dataset, capable of detecting various objects on which the YOLO-NAS model has been trained.

DeepSort, or Deep Association Metric SORT, is a tracking algorithm that tracks detected objects across multiple frames, leveraging their spatial and temporal characteristics. DeepSort employs a neural network for feature extraction and an association metric for matching detections with tracked objects.

Together, YOLO-NAS and DeepSort ensure efficient people tracking in videos. Detections generated by YOLO-NAS models are passed to DeepSort, which then tracks the objects.

SlowFast is a neural network architecture designed for video classification, consisting of two streams: a *slow* stream processing low-frame-rate information, focusing on high-level semantic details, and a *fast* stream processing high-frame-rate information, focusing on low-level temporal details. This combination allows SlowFast to classify videos effectively by capturing long-term and short-term details.

The SlowFast architecture is well-suited for video activity classification as it can capture and analyze spatial and temporal characteristics. Its dual-stream structure enables simultaneous processing and synthesis of information regarding long-term and short-term changes in videos, making it an ideal choice for tasks requiring an understanding of complex and dynamic activities.

ResNet-50 is often used as the backbone for SlowFast. It is a deep convolutional neural network designed for image classification, featuring residual blocks that allow efficient learning on large datasets and extraction of complex features from images. This makes ResNet-50 an excellent choice as a backbone in SlowFast, as it can effectively process and analyze both spatial and temporal characteristics of videos.

3. Experiment and Results

Let us formulate an algorithm for solving the shoplifting detection task using the SlowFast neural network. The algorithm can be divided into four main stages:

- **Stage 1.** Data Collection:

In the first stage, a dataset is gathered for subsequent use in training and testing the model. In our case, the dataset consists of video clips capturing instances of shoplifting from surveillance cameras in retail stores. The dataset was created manually, meaning a sufficient amount of video footage was recorded on surveillance cameras using the services of actors.

- **Stage 2.** Data Processing:

a) Video segments with each individual are extracted from raw video using computer vision algorithms. YOLO-NAS is employed for person detection, and DeepSort is utilized for tracking specific individuals across frames.

b) The frame dimensions are standardized to (256, 256, 3), where 256 represents the height and width of the frame, and 3 corresponds to the RGB colour model.

c) Extracted video segments are divided into shorter segments (3-5 seconds), labeled with classes such as "shoplifting" – 1 and "non-shoplifting" – 0. The dataset is split into training (14 084), validation (3 521), and test (1 957) sets.

d) Artificial data augmentation techniques are applied to increase the amount of data in the training set. The underrepresented class was over-sampled to address the problem of unbalanced data in the dataset, given that all samples related to a particular image (original and augmented versions) were placed in the same part of the dataset.

- **Stage 3.** Fine-Tuning:

The third stage involves fine-tuning the pre-trained SlowFast model on the Kinetics-400 dataset. This process includes precise adjustment on the training set and hyperparameter tuning on the validation set.

- **Stage 4.** Model Integration:

The fourth and final stage is the integration of the model into a product. This involves applying techniques such as quantization and pruning to optimize the model for deployment on devices with limited computational power.

Upon completion of this algorithm, a model will be constructed and prepared for deployment, capable of accurately detecting shoplifting in video footage.

Figure 1 demonstrates a moment of theft from a randomly selected video sample.

Figure 1. Example of an image from the dataset

The values of calculated accuracy, recall, and F1 scores for each of the classes are presented in Table 1.

Table 1. Values of accuracy, precision, recall and F1-score indicators of the experiment

	accuracy	**precision**	**recall**	**F1-score**
Not Shoplifting	0.95	0.93	0.97	0.95
Shoplifting	0.95	0.96	0.93	0.95

Thus, the accuracy of the conducted classification is 95%. Since the sample

was balanced, considering the presented values of the metrics, this value sufficiently and fully characterizes the classification result.

4. Conclusion

This research aimed to establish an end-to-end pipeline for identifying shoplifting incidents using video data obtained from security cameras. Several neural networks and algorithms, namely YOLO-NAS, DeepSort, and SlowFast, were integrated to accomplish this goal. Additionally, a neural classifier for activity recognition was fine-tuned specifically for the shoplifting detection task.

The dataset used in this study was captured under real-world conditions with the participation of actors. The classifier demonstrated a notable performance across all metrics, achieving an accuracy of approximately 95%, surpassing the accuracy of models presented in previous research by several percentage points. It is also important to note that the datasets are different, but the current one is more representative in terms of real data, as it includes a large variation of all kinds of theft cases. Moreover, the trained classifier exhibited high efficiency, making it suitable for deployment in real-time applications. Subsequent research endeavors will concentrate on the practical implementation of the proposed model in shopping malls.

References

[1] Islam, M. S., Sultana, S., Roy, U. K., and Mahmud, J. A. A review on Video Classification with Methods, Findings, Performance, Challenges, Limitations and Future Work. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 6 (2): pages 47-57, 2021. doi: 10.26555/jiteki.v6i2.18978.

[2] Nasaruddin, N., Muchtar, K., Afdhal, A., and Dwiyantoro, A. P. J. Deep anomaly detection through visual attention in surveillance videos. *Journal of Big Data*, 7(1): 87, 2020, doi: 10.1186/s40537-020-00365-y.

[3] Ansari, M. A. and Singh D. K. ESAR, An Expert Shoplifting Activity Recognition System. *Cybernetics and Information Technologies*, 22(1): pages 190–200, 2022. doi: 10.2478/cait-2022-0012.

[4] Kirichenko, L., Radivilova, T., Sydorenko B., and Yakovlev S. Detection of Shoplifting on Video Using a Hybrid Network. *Computation*, 10(11): 199, 2022. doi: 10.3390/computation10110199.

[5] Terven, J., Córdova-Esparza, D.-M., and Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4): pages 1680-1716, 2023, doi: 10.3390/make5040083.

[6] Feichtenhofer, C., Fan, H., Malik, J., and He, K. SlowFast Networks for Video Recognition. In *the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210. IEEE Computer Society, 2019. doi: 10.1109/ICCV.2019.00630.

[7] Aharon, S. et al. Deci-AI/super-gradients: 3.6.0. 2024. doi: 10.5281/ZENODO.10568661.

[8] Aharon, S. et al. Deci-AI/super-gradients: 3.0.8. 2023. doi: 10.5281/ZENODO.7789328.

[9] Ahmad, S. Object Tracking with DeepSORT. 2024. URL https://medium.com/@shoebahmad_AI/object-tracking-with-deepsort-by-shoeb-ahmad-53d9c9e22f99

A fast prediction of the OCR results based on binary image quality assessment methods

Mateusz Kopytek, Krzysztof Okarma[0000−0002−6721−3241]

West Pomeranian University of Technology in Szczecin
Department of Signal Processing and Multimedia Engineering
26 Kwietnia 10, 71-126 Szczecin, Poland
{km46880, okarma}@zut.edu.pl

Abstract. *Since the first step of the Optical Character Recognition (OCR) procedure is usually image binarization and the execution of the OCR process takes longer time than image quality assessment (IQA) of binary images, a fast prediction of the OCR results before text recognition would be interesting for non-uniformly illuminated document images, also for applications in mobile devices. Since IQA methods for binary images usually require the presence of a reference image, an important problem is to avoid this necessity as proposed in the paper. Experimental results, obtained by applying mutual comparisons for various binarization methods, are promising and confirm the validity of the proposed approach.*
Keywords: *Optical Character Recognition, document images, image quality assessment, image binarization*

1. Introduction

Rapid progress in the development of artificial intelligence methods has revolutionized many areas of science and technology. One of them is undoubtedly computer vision and image processing. However, there are still some applications where AI-based solutions are not always the best choice, e.g., due to well-known overfitting problem as well as the lack of explainability. Nevertheless, one of the most widely known application of such methods is the Optical Character Recognition (OCR), where many models have been trained, particularly considering the dictionary and language-related modules. A good example of such AI-based software is the well-known and popular open-source Tesseract OCR engine, initially developed by Hewlett Packard and then sponsored by Google.

DOI:10.17388/WUT.2024.0002.MiNI

Nevertheless, in the presence of noise and some other artifacts, particularly when combined with non-uniform illumination, even with an improper image preprocessing, the obtained OCR results might be unsatisfactory. Therefore, it would be advantageous to reliably predict the OCR accuracy for such images, especially if another image could be captured quickly. Hence, the idea of the paper is to validate the usefulness of certain fast binary image quality assessment (IQA) methods for this purpose, thus avoiding the need to run the full OCR procedure. Such an approach could be particularly useful for mobile devices equipped with a camera, such as smartphones, which are often used for capturing document images.

2. The motivation and the proposed approach

The OCR accuracy is dependent on several factors, such as the quality of applied software, availability of dictionaries and language models, as well as the input image quality [1, 2]. The correctness of the text recognition results may be expressed in several ways, however, one of the most popular is the edit distance, known as Levenshtein distance, between two text strings, defined as the number of single-character text edits (insertions, deletions or substitutions) necessary to convert one string into the other. Knowing the "ground-truth" text of the document presented on the recognized image, it is possible to calculate it directly after the end of the OCR process. To compare the results for various image binarization algorithms, applied instead of the default first step of the OCR algorithm, as well as the influence of image quality, some document image datasets have been prepared. One of the most recent ones[1], containing 176 non-uniformly illuminated document images with commonly used placeholder text "Lorem ipsum" captured by Digital Single Lens Reflex (DSLR) camera Nikon N70, known as WEZUT OCR Dataset [3], has been used in the paper to verify the appropriateness of the proposed approach.

One of the potential approaches to prediction of the OCR results, in our case defined by the Levenshtein distance (LD), might the use of no-reference (NR) quality assessment of input images. Although there are many such metrics [4], also AI-based, it should be noted that the text recognition is performed for binary images. Therefore, the quality of the input image and the binary image being the direct input for the OCR procedure may differ depending on the binarization method, particularly for non-uniformly illuminated images.

[1]Dataset available at: `https://okarma.zut.edu.pl/?id=dataset`

Although the application of the NR IQA metrics seems to be obvious, an information related to the quality of binary images would be more important. Unfortunately, the quality assessment of binary images is usually conducted by comparing pixels of the assessed image to the respective pixels from the "ground-truth" (GT) binary image. Hence, such metrics cannot be considered as "blind" (no-reference), since the reference image is necessary, and it is not always available.

To solve this problem, a mutual comparison of two binary images, obtained as the results of two different binarization methods applied to the same image, is proposed. Assuming the poor quality of the binary image, e.g., caused by the use of a simple global thresholding, such as the Otsu method [5], for a non-uniformly illuminated image, the differences between such an image and others achieved using different binarization algorithms are expected to be relatively high. On the other hand, relatively high similarity in binarization results should be typical rather for high quality images and good OCR results might be expected for both binary images.

In one of the earlier papers [6], a simplified method for predicting OCR results was considered based on the combined assessment of binarization results. Apart from assessing the text recognition results using recognition accuracy without the use of Levenshtein distance, the main difference between this method and the proposed approach lies in the use of the combined binary image quality metric based on the comparison with the "ground-truth" image (a full-reference IQA approach). Hence, it was assumed that such an image is available and geometrically matched, i.e., no additional image registration is needed. Additionally, only three global binarization methods were considered, and the experiments were conducted using a limited number of images. As stated in the conclusions [6], further research should concentrate on developing some no-reference metrics as proposed in the paper.

3. The verification and obtained results

The proposed idea was verified using 176 images from the WEZUT OCR Dataset, which represent photos of documents printed in 5 different popular font styles (Arial, Times New Roman, Calibri, Courier, and Verdana) w with typical attribute modifications (normal, bold, italic, and bold italics). The dataset is primarily developed mainly for preprocessing non-uniformly illuminated document images, preparing them for further text recognition using various OCR engines.

Table 1. Pearson's correlation of individual IQA metrics with the Levenshtein distance for the considered 1584 images

Quality metric	**PCC**
Distance Reciprocal Distortion (DRD) [14]	0.3359
Misclassification Penalty Metric (MPM) [15]	0.1548
Peak Signal-to-Noise Ratio (PSNR)	0.2434
Precision	0.3728
Recall	0.0508
F-Measure	**0.8146**
pseudo-Precision	0.4190
pseudo-Recall	0.0503
pseudo-F-Measure	**0.8574**
Sensitivity	0.0508
Specificity	0.0615
Balanced Classification Rate (BCR)	0.0452
S-F-Measure	0.1184
Accuracy	0.2365
Geometric Accuracy	0.1289
Negative Rate Metric (NRM)	0.0452
Border Distance (chessboard) [16]	0.2487
Border Distance (city-block) [16]	0.2318
Border Distance (Euclidean) [16]	0.2477

To assess the applicability of different binary IQA methods, the dataset was expanded by applying various binarization techniques, resulting in a total of 1584 images. Each image underwent the following binarization methods: Bernsen [7], Bradley [8], Liedes (based on deep learning)[2], Feng [9], Meanthresh, NICK [10], Sauvola [11], Wolf [12] and Niblack [13].

The text recognition was conducted using Tesseract OCR engine without dictionary and language support to prevent the influence of such additional modules on the obtained results.

For each of the original images, the results obtained using the aforementioned 9 binarization methods were compared with the remaining 8 binary images using various binary image quality metrics. Consequently, $\frac{8 \cdot 9}{2} = 36$ calculations of each

[2]The code is available at: `https://github.com/sliedes/binarize`

metric should be made. Then, the average mutual quality scores were calculated from these 8 results. Finally, 1584 average mutual quality scores were obtained for each metric and the Pearson's correlation coefficients between each of these metrics and the previously determined Levenshtein distances were computed. The correlation values obtained with the OCR results for various metrics are presented in Table 1. As shown, many metrics based on the counts of true positive, true negative, false positive, and false negative pixels demonstrated relatively low correlation with the OCR results. On the other hand, the undoubtedly best results were obtained for the F-Measure and pseudo-F-Measure metrics.

4. Conclusions and future work

The presented results confirm the validity of the proposed approach, particularly when applying F-Measure or pseudo-F-Measure as the mutual binary image similarity metrics, which are highly correlated with the OCR results achieved for respective binary images. Subjectively, the best binarization results for the considered WEZUT OCR Dataset have been observed for the deep learning based method developed by Sami Liedes, further leading to the lowest values of the Levenshtein distances for all images. Therefore, our further research may concentrate on the use of other deep learning-based image binarization methods, also considering their combination using the voting approach. Nevertheless, as stated in the paper [17], "*DL-based binarization solutions are very sensitive to logical text size and they are still not effective in this domain, thus requiring major improvements*".

The use of an additional image binarization method does not significantly influence the obtained correlation for the selected binary IQA metric, as worse methods producing poor quality results should lead to relatively high values of the Levenshtein distance, whereas better binarization methods should lead to better OCR results. Such a relationship follows the results obtained for the considered adaptive image binarization methods, thus a similarly good prediction may be expected for some other binarization algorithms as well.

On the other hand, the combination of various binary image quality assessment methods, including the use of shallow neural networks, appears to be a natural direction for further experiments aimed at achieving even higher correlation with OCR accuracy. This correlation can also be verified with other similar available datasets, including video sequences.

References

[1] Bernardino, R., Lins, R. D., and da Silva Barboza, R. A quality, size and time assessment of the binarization of documents photographed by smartphones. *Journal of Imaging*, 9(2):41, 2023. doi:10.3390/jimaging9020041.

[2] Lins, R. D., Bernardino, R. B., da Silva Barboza, R., and Simske, S. J. Binarization of photographed documents image quality, processing time and size assessment. In *Proceedings of the 22nd ACM Symposium on Document Engineering*. ACM, San Jose, CA, USA, 2022. doi:10.1145/3558100.3564159.

[3] Michalak, H. and Okarma, K. Robust combined binarization method of non-uniformly illuminated document images for alphanumerical character recognition. *Sensors*, 20(10), 2020. doi:10.3390/s20102914. Article no. 2914.

[4] Kamble, V. and Bhurchandi, K. No-reference image quality assessment algorithms: A survey. *Optik*, 126(11–12):1090–1097, 2015. ISSN 0030-4026. doi:10.1016/j.ijleo.2015.02.093.

[5] Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. ISSN 0018-9472, 2168-2909. doi:10.1109/TSMC.1979.4310076.

[6] Lech, P. and Okarma, K. Prediction of the optical character recognition accuracy based on the combined assessment of image binarization results. *Elektronika ir Elektrotechnika*, 21(6), 2015. ISSN 1392-1215. doi: 10.5755/j01.eee.21.6.13764.

[7] Bernsen, J. Dynamic thresholding of gray level image. In *ICPR'86 Proceedings of International Conference on Pattern Recognition*, pages 1251–1255. Berlin, Germany, 1986.

[8] Bradley, D. and Roth, G. Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools*, 12(2):13–21, 2007. ISSN 1086-7651. doi:10.1080/2151237X.2007.10129236.

[9] Feng, M.-L. and Tan, Y.-P. Contrast adaptive binarization of low quality document images. *IEICE Electronics Express*, 1(16):501–506, 2004. ISSN 1349-2543. doi:10.1587/elex.1.501.

[10] Khurshid, K., Siddiqi, I., Faure, C., and Vincent, N. Comparison of Niblack inspired binarization methods for ancient documents. In *Document Recognition and Retrieval XVI*, volume 7247, pages 7247 – 7247 – 9. 2009. doi: 10.1117/12.805827.

[11] Sauvola, J. and Pietikäinen, M. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000. ISSN 00313203. doi:10.1016/S0031-3203(99)00055-2.

[12] Wolf, C. and Jolion, J.-M. Extraction and recognition of artificial text in multimedia documents. *Formal Pattern Analysis & Applications*, 6(4), 2004. ISSN 1433-7541, 1433-755X. doi:10.1007/s10044-003-0197-7.

[13] Niblack, W. *An Introduction to Digital Image Processing*. Prentice-Hall, Inc., USA, 1990. ISBN 0-13-480674-3.

[14] Lu, H., Kot, A., and Shi, Y. Distance-reciprocal distortion measure for binary document images. *IEEE Signal Processing Lett.*, 11(2):228–231, 2004. doi: 10.1109/lsp.2003.821748.

[15] Young, D. and Ferryman, J. PETS metrics: On-line performance evaluation service. In *Proc. 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 317–324. IEEE, Beijing, China, 2005. doi:10.1109/vspets.2005.1570931.

[16] Zhang, F., Cao, K., and Zhang, J.-L. A simple quality evaluation method of binary images based on border distance. *Optik*, 122(14):1236–1239, 2011. doi:10.1016/j.ijleo.2010.07.030.

[17] Sánchez-Rivero, R., Bezmaternykh, P., Morales-González, A., Silva-Mata, F. J., and Bulatov, K. Assessing the relationship between binarization and OCR in the context of deep learning-based ID document analysis. In *Progress in Artificial Intelligence and Pattern Recognition*, pages 134–144. Springer International Publishing, 2021. doi:10.1007/978-3-030-89691-1_14.

Comparison of deep learning approaches to violence detection in videos

Filip Adamiok, Artur Wilkowski[0000−0002−6814−7645]

Warsaw University of Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19 00-665 Warsaw, Poland
artur.wilkowski@pw.edu.pl

Abstract. *The paper compares six different types of deep learning architectures in the problem of detecting violent behavior in videos. The evaluated architectures are 3DCNN, ConvLSTM, LRCN, Xception+LSTM, S3D and MViTv2, which cover broad range of network architectural concepts: 3D convolutions, recurrent networks and transformers. Additionally, there is assessed an extent to which transfer learning approach can help to boost classifiers' performance. Two datasets UCF-Crime and RWF-2000 are used for benchmark. It is shown that the modern transformer architecture MViTv2 can show superior results, especially when coupled with careful selection of training data and a multi-stage application of transfer learning.*
Keywords: *violence detection in videos, human action recognition, transformer*

1. Introduction

Nowadays, our urban environment is filled with surveillance cameras. These cameras provide footage that help appropriate security services to quickly react to unlawful or dangerous situations but also assist in investigations or provide evidence. A broad category of dangerous situations include violent behaviors. Monitoring by a human operator of such data is tedious due to the large area, abundance of sensors and relative rarity of violent situations. Therefore, there is an urgent need for automated system that would help to filter video materials and focus operator's attention on the most suspicious fragments.

DOI:10.17388/WUT.2024.0002.MiNI

In this paper there are evaluated 6 deep learning concepts often used in the problem of Human Action Recognition and their applicability to violence detection. There is also analyzed an extent to which transfer learning approach can help to improve classification results of video clips. It is shown that some of the evaluated configurations compares favorably to the existing state-of-the-art methods for the problem discussed.

2. Evaluated network architectures

Six different network architectures were evaluated for the problem. First three networks were trained from scratch, remaining networks used transfer learning in the form of pre-training on ImageNet or Kinetics400 dataset.

2.1. Models trained from scratch

3DCNN. The network is based on 3D convolutions instead of 2D convolutions in order to capture both spatial and temporal structure of the video material. The network structure is given Table 1. In the network 3D convolutions are followed by 3D maximum pooling operations to reduce spatial and/or temporal dimension while increasing the number of channels. Dropouts are added to reduce overfitting. The top of the network consists of fully connected layers to compute the final result.

ConvLSTM. ConvLSTM [1] is an extension to the original LSTM specification. The main difference is that 1D tensors that LSTM operates on are replaced with 3D tensors (as in the classical CNN networks) and fully connected layers present in an LSTM cell are replaced with convolutions. Thanks to this the ConvLSTM cell is better suited to process spatio-temporal data than original LSTM cell. The proposed network structure based on the ConvLSTM concept is provided in Table 2.

LRCN. The concept of network structure was introduced in [2]. The network combines feature extraction CNN layers applied to single image frames with LSTM layers to capture temporal relations. The general structure of the proposed network based on this concept is given in Table 4. After applying the CNN layers (Table 3) as well as LSTM layers, fully connected layers are used for final classification.

Table 1: 3DCNN

Filter type	params.
Conv3D	$(2, 2, 2) \times 64$
MaxPool3D	(1,2,2)
Dropout	20%
Conv3D	$(2, 2, 2) \times 128$
MaxPool3D	(1,2,2)
Dropout	20%
Conv3D	$(2, 2, 2) \times 128$
MaxPool3D	(2,2,2)
Dropout	20%
Conv3D	$(2, 2, 2) \times 256$
MaxPool3D	(2,2,2)
Dropout	20%
Conv3D	$(2, 2, 2) \times 512$
MaxPool3D	(2,2,2)
Dropout	20%
Flatten	-
FullyConn	1028
Dropout	20%
FullyConn	1028
Dropout	20%
FullyConn	256
FullyConn	2

Table 2: ConvLSTM

Filter type	params.
ConvLSTM	$(3, 3) \times 4$
MaxPool3D	(1,2,2)
Dropout	20%
ConvLSTM	$(3, 3) \times 8$
MaxPool3D	(1,2,2)
Dropout	20%
ConvLSTM	$(3, 3) \times 14$
MaxPool3D	(1,2,2)
Dropout	20%
ConvLSTM	$(3, 3) \times 16$
MaxPool3D	(1,2,2)
Dropout	20%
Flatten	-
FullyConn	1028
FullyConn	1028
FullyConn	256
FullyConn	2

Table 3: CNN backbone of LRCN

Filter type	params.
Conv2D	$(3, 3) \times 32$
MaxPool2D	(2,2)
Conv2D	$(3, 3) \times 64$
MaxPool2D	(2,2)
Conv2D	$(3, 3) \times 128$
MaxPool2D	(2,2)
Conv2D	$(3, 3) \times 256$
MaxPool2D	(2,2)
Conv2D	$(3, 3) \times 512$
MaxPool2D	(2,2)
Conv2D	$(3, 3) \times 1028$
MaxPool2D	(2,2)
GlbMaxPool2D	-

Table 4: LRCN

Filter type	params.
CNN backbone	-
LSTM	128
LSTM	128
FullyConn	1028
Dropout	20%
FullyConn	1028
Dropout	20%
FullyConn	256
FullyConn	2

Table 5: Xception+LSTM

Filter type	params.
Xception	(Imagenet)
LSTM	32
LSTM	32
FullyConn	2048
Dropout	20%
FullyConn	1028
Dropout	20%
FullyConn	32
FullyConn	2

2.2. Models utilizing transfer learning

Xception+LSTM. In order to leverage feature extraction capabilities of pre-trained models, a transfer-learning approach was adopted by combining the Xception network [3] pre-trained on the ImageNet dataset with LSTM layers to capture temporal relations. The network architecture follows the concept of LRCN. In the proposed network structure, given in Table 5, image features are extracted by the Xception module and then further processed by LSTM cells and fully connected layers.

S3D. This is an extension of 3DCNN modified so to reduce the number of parameters without significant reduction of network expressiveness. In order to do so two important improvements were proposed in [4]. Firstly, initial 3D convolutions are replaced with 2D convolutions, than 2D spatial convolutions are mixed with 1D temporal convolutions to reduce parameter set. In the experiments third party implementation of S3N pre-trained on Kinetics400 dataset was used.

MViTv2. In contrast to earlier applications of transformer networks which maintain data volume shapes through processing, the Multi-Scale Vision Transformer uses a 'multiscale vision hierarchies' principle to process data. Initially, for low-level feature extraction, a fine-grained processing (characterized by large spatio-temporal volumes with small channel depth) is applied, then a coarser-grained processing is utilized with smaller spatio-temporal volumes and larger channel depth. In the experiments an improved version of the concept MViTv2 [5] pre-trained on Kinetics400 dataset was used.

3. Datasets and preprocessing

Two datasets were used in the process of model training and evaluation: UCF-Crime [6] containing 1900 videos (128 hours) and RWF-2000 [7] containing 2000 videos (5 sec. each). Both datasets included a wide array of behaviors categorized as violent, such as fighting, robbery, shootings, and assault. The UCF-Crime dataset was preprocessed before use. The preprocessing steps consisted of:

- formation of two new classes - 'violent behavior' and 'non-violent behavior'
- splitting videos into 5-sec. clips and assigning each clip to the appropriate class

As a result, more balanced dataset of 930 video clips was obtained with much less teacher errors resulting from the fact that in the original dataset the actual violent action often spanned only a small portion of the video.

Typical length of a clip was about 5 seconds in both datasets. Each sequence was uniformly sampled to obtain a tensor of dimensions $(T, W, H, 3)$ with symbols denoting (T)ime span, image (W)idth, (H)eight.

4. Experiments

For models not utilizing transfer learning, the following hyperparameters were optimized using grid-search: number of fully connected layers, optimizer type (SGD, ADAM), input volume size parameters T, W, H, learning rate, learning rate reduction patience, early stopping patience and batch size. In contrast, for models using transfer learning the parameters W and H were fixed to 224, T was fixed to 16 and additional tuned hyper-parameter was the number of frozen layers in the backbone network. Cross-entropy loss was used in the training of all studied models.

4.1. UCF-Crime dataset

The dataset was divided into training and validation dataset (no testing dataset) in proportions 80/20. The performance of models are given in Table 6.

Table 6: Results for the UCF-Crime dataset

model	no. params.	batch	opt.	loss	acc.	TNR	TPR	F1
3DCNN	54,413,658	5	SGD	0.5	75.0%	0.70	0.78	0.78
ConvLSTM	45,648,874	10	ADAM	0.6	72.2%	0.80	0.66	0.73
LRCN	8,484,454	5	SGD	0.57	73.3%	0.78	0.69	0.74
Xception + LSTM	23,420,842	5	SGD	0.55	74.4%	0.68	0.79	0.78
S3D	**7,912,098**	5	SGD	0.34	82.4%	0.90	0.76	0.83
MViT	34,231,682	2	SGD	**0.23**	**92.6%**	**0.95**	**0.91**	**0.93**

The best results were obtained by the MViTv2 model with a large margin over S3D, which however achieve good results with a minimum number of parameters. Models trained from scratch performed generally worse than model using knowledge transfer, but a margin between LRCN and Xception + LSTM was shown to be very narrow.

4.2. RWF-2000 dataset

The final experiment consisted of applying the best model MViTv2 to the RWF-2000 dataset. In the first 'cross-dataset' scenario there was evaluated how the model trained on UCF-Crime performs on RWF-2000 dataset (model MViT-UCF). In the second scenario the Kinetics400-pre-trained model was directly tuned on RWF-2000 dataset (model MViT-RWF). In the last scenario, the Kinetics400-pre-trained model was first tuned on UCF-Crime and then on RWF-2000 dataset (model MViT-UCF+RWF). For the RWF-2000 dataset there was used an 80/20 train/test split proposed by the dataset authors. The training set was further split in proportions 80/20 for the training/validation procedure. Results of three models compared to other state-of-the-art results are given in Table 7.

Table 7: Results for the RWF-2000 dataset

model	**acc.**
3DCNN [7]	87.25%
SepConvLSTM-M [8]	89.75%
CNN(VGG16)+ConvLSTM [9]	92.40%
VDSTR [10]	93.80%
Structured keypoint pooling [11]	93.40%
MViT-UCF (ours)	85.75%
MViT-RWF (ours)	93.00%
MViT-UCF+RWF (ours)	94.25%
3DCNN + CBAM on Grid Frames [12]	**99.20%**

The best of our models is the model using extensively transfer learning and trained on both UCF-Crime and RWF-2000 datasets (achieving accuracy of 94.25%) and is better than the one trained only on RWF-2000 (93.00%). The results also compare favorably to other state-of-the-art methods and is second only to the most recent model [12] (published on December '23) which was developed in parallel to our works. It can be also observed that the model trained in 'cross-dataset' scenario also provides decent results (85.75%) which is comparable to the performance of earlier convolutional networks.

5. Conclusions

Three basic conclusions can be drawn from the experiments carried out. First of all, according to expectations, it was shown that the best performance can be

obtained using a modern transformer network MViTv2, despite the large number of parameters to be trained. Its results compares well to the current state-of-the-art on the RWF-2000 dataset. Secondly, it was confirmed that non-transfer learning models generally perform worse that their counterparts utilizing TL. Thirdly, it was also shown that multi-stage tuning on different datasets (Kinetics400 → UCF-Crime → RWF-2000) can also provide boost in performance.

References

[1] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. Convolutional lstm network: a machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 802–810. MIT Press, Cambridge, MA, USA, 2015.

[2] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017. doi: 10.1109/TPAMI.2016.2599174.

[3] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807. IEEE Computer Society, Los Alamitos, CA, USA, 2017. ISSN 1063-6919. doi:10.1109/CVPR.2017.195.

[4] Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 318–335. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01267-0.

[5] Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., and Feichtenhofer, C. Mvitv2: Improved multiscale vision transformers for classification and detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4794–4804. 2022. doi: 10.1109/CVPR52688.2022.00476.

[6] Sultani, W., Chen, C., and Shah, M. Real-world anomaly detection in surveillance videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488. 2018. doi:10.1109/CVPR.2018.00678.

[7] Cheng, M., Cai, K., and Li, M. Rwf-2000: An open large scale video database for violence detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4183–4190. IEEE Computer Society, Los Alamitos, CA, USA, 2021. ISSN 1051-4651. doi:10.1109/ICPR48806.2021.9412502.

[8] Islam, Z., Rukonuzzaman, M., Ahmed, R., Kabir, M. H., and Farazi, M. Efficient two-stream network for violence detection using separable convolutional lstm. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. 2021. doi:10.1109/IJCNN52387.2021.9534280.

[9] Mugunga, I., Dong, J., Rigall, E., Guo, S., Madessa, A. H., and Nawaz, H. S. A frame-based feature model for violence detection from surveillance cameras using convlstm network. In *2021 6th International Conference on Image, Vision and Computing (ICIVC)*, pages 55–60. 2021. doi:10.1109/ICIVC52351.2021.9526948.

[10] Chelali, M., Kurtz, C., and Vincent, N. Violence detection from video under 2d spatio-temporal representations. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2593–2597. 2021. doi:10.1109/ICIP42928.2021.9506142.

[11] Hachiuma, R., Sato, F., and Sekii, T. Unified keypoint-based action recognition framework via structured keypoint pooling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22962–22971. 2023.

[12] Hwang, I.-C. and Kang, H.-S. Anomaly detection based on a 3d convolutional neural network combining convolutional block attention module using merged frames. *Sensors*, 23(23):9616, 2023.

Chapter 7

Uncertainty in Artificial Intelligence

Track Chairs:

- prof. Beata Zielosko - University of Silesia in Katowice
- prof. Agnieszka Jastrzębska - Warsaw University of Technology
- dr Piotr Wasilewski - Systems Research Institute of the Polish Academy of Sciences

From three-valued logic to balanced fuzzy logic - concepts vs. possible applicability

Zofia Matusiewicz[1][0000−0003−0523−0983]

[1]*University of Information Technology and Management*
Chair of Cognitive Science and Mathematical Modelling
ul. Sucharskiego 2, 35-225 Rzeszow, Poland
zmatusiewicz@wsiz.edu.pl

Abstract. *This article presents balanced fuzzy logic as an extension of both the idea of three-valued logic and fuzzy logic. In balanced fuzzy logic, three states are also considered. The first one denotes a true sentence. Conviction of truthfulness can be incomplete; therefore, in this case, the degree of certainty is a value in the interval* $(0, 1]$*, a sentence about which we are not able to say whether it is true nor false, denoted by* 0*, and a sentence about which we can say that it is false given certain information. The degree of determination of falsity of a sentence is a value in the interval* $[-1, 0)$*.*
Keywords: *three-valued logic, fuzzy balanced logic, uncertainty*

1. Introduction

In the late 1920s, Jan Lukasiewicz introduces three-valued logic [1]. Let us first recall the basic concepts and assumptions of three-valued logic. It assumes that sentences can take a third logical value - the so-called possibility- in addition to true and false. The introduction of the value of $\frac{1}{2}$ for those times becomes an intensely debated issue, giving rise to much controversy. Mathematicians of the time, such as Ajdukiewicz, Borkowski, Leśniewski, Smólka, Twardowski, have many doubts about the interpretation of the introduced value of $\frac{1}{2}$ (see [2], [3]).

However, suppose we assume that this third value represents the value of an expression about which there is uncertainty about its truth or a lack of information to determine whether a given sentence is true or false. In that case, this kind of three-valued logic seems to be directly related to IT-related problems related to data analysis. Such an interpretation goes somewhat beyond the epistemological and ontological interpretations considered in philosophy.

DOI:10.17388/WUT.2024.0002.MiNI

2. Three-valued and balanced fuzzy logic

In 2006, Homenda introduced the concept of balanced fuzzy sets in which the degree of membership is bipolar [4], [5]. Introducing the balanced fuzzy sets opens up possibilities for us, that is, the simultaneous consideration of positive and negative data in the decision-making process, among other things [6], [7]. So, balanced fuzzy logic leads us to the possibility of considering true and false in some degree sentences. Moreover, it provides an opportunity to describe a state where it is difficult to judge the truth of a sentence.

2.1. Some similarity between three-valued logic and balanced fuzzy logic

Note that if we replaced the Łukasiewicz values $(0, \frac{1}{2}, 1)$ corresponding to linguistic values (true, possibility, false) with $(1, 0, -1)$ corresponding to their meanings (true, uncertainty, falseness), then the further process by which balanced fuzzy logic is created is rather obvious. The similarity between the idea of three-valued logic (3VL), its modification (3VL-B) and balanced fuzzy logic (BFS) is presented in Table 1. In balanced fuzzy logic, the truth and falseness of sentences is a fuzzy

Table 1. Values in 3VL, 3VL-B, BFS

state	**3VL**	**3VL-B**	**BFS**
True	1	1	$(0, 1]$
Possibility/Uncertainty	$\frac{1}{2}$	0	0
False	0	-1	$[-1, 0)$

value. If there is no doubt that a statement is true, we assign it a value of 1. We can define the certainty function of a sentence in various ways. It may be determined on the basis of knowing the truth of the premises allowing one to assess the truth of a sentence. Similarly, a statement is false when it is assigned the value -1. If not all the premises allowing the statement to be fully false or true, then we assign it a value in the range $[-1.0)$. However, there are sentences whose truth or falsity is difficult to assess with only incomplete data or no data at all (see Figure 1).

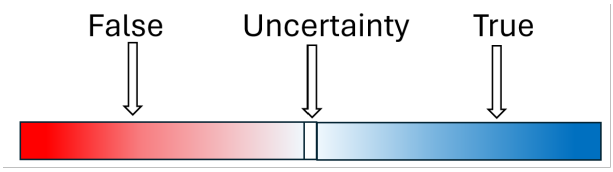


Figure 1. Concept of values of BFL

2.2. Knowledge representation in 3VL-B and in BFL

The main issue of this sections is the possibility of representing knowledge representations by using 3VL and BFL. We use a tabular representation of knowledge for this purpose (as proposed in [8]). Such tabular representation of data is also called information systems or attribute-value systems. A knowledge representation system can be viewed as a table of data whose columns are marked with attributes, rows are marked with objects (states, processes, etc.), and each row represents information about the corresponding object. In our approach, we propose describing features with values determining the degree to which the feature is fulfilled.

According to the principle specified in Table 1, decision systems from 3VL-B (Table 2) and BFL (Table 3) describing people at risk of heart attack will be presented. In Table 3, the true and false values were determined according to the formula 1.

$$l(x) = \begin{cases} \frac{|x|}{|\max(U_a)|}, & for\ x\ that\ fulfils\ the\ attribute \\ 0 & lack\ of\ data\ or\ not\ fulfiling\ the\ given\ or\ opposite\ attribute \\ \frac{-|x|}{|\min(U'_a)|}, & for\ x\ fulfils\ the\ opposite\ attribute \end{cases}, \tag{1}$$

where U_a is whole set of possible values and U'_a negative values. We use the following notations: **HC**:High cholesterol, **HBP**: High blood pressure, **D**: Diabetes, **O**: Obesity, **S**: Smoking, **PI**: Physical inactivity. These features were selected according to the basic medical description of the heart attack risk problem (for example [9].

To clarify, in the same way that the membership function for fuzzy sets is defined, the truth and falsity functions of a given statement should be defined for BFL. Of course, the issue is quite wide and cannot be reduced to simply counting the premises for and against a given statement, but also based on expert knowledge

Table 2. Example of 3VL-B decision systems describing people at risk of heart attack.

	HC	**HBP**	**D**	**O**	**S**	**PI**
Patient 1	1	1	0	1	1	0
Patient 2	1	−1	1	0	−1	−1
...	...	...	...	...	...	...
Patient (n-1)	1	−1	0	0	−1	1
Patient n	0	0	1	1	1	−1

Table 3. Example of BFL decision systems describing people at risk of heart attack.

	HC	**HBP**	**D**	**O**	**S**	**PI**
Patient 1	0.6	0.78	0	0.9	1	0
Patient 2	0.82	−0.45	0.56	0	−0.89	−0.4
...	...	...	...	...	...	...
Patient (n-1)	0.8	−0.5	0	0	−0.2	0.6
Patient n	0	0	0.7	0.63	0.79	−0.23

and other methods.

3. Conclusions

Balanced fuzzy sets and logic are an extension of trivalent logic. At the same time, it is also an extension of sets and fuzzy logic. Combining these two properties, we observe that this concept can be used to represent incomplete and uncertain knowledge.

Acknowledgment

The work was supported by grant: "Fuzzy balanced sets in the development of real-world models", University of Information Technology and Management, Poland.

References

[1] Łukasiewicz, J. O logice trójwartościowej. *Ruch Filozoficzny*, (5(9)):170a–171a, 1920.

[2] Łukasiewicz, J., Smolka, F., and Leśniewski, S. U źródeł logiki trójwartościowej. *Filozofia Nauki*, 2(3-4):227–240, 1994.

[3] Garbacz, P. Filozoficzne motywacje logiki trójwartościowej Jana Łukasiewicza. *Roczniki Filozoficzne*, pages 75–100, 1997.

[4] Homenda, W. Balanced fuzzy sets. *Information Sciences*, 176(17):2467–2506, 2006.

[5] Matusiewicz, Z. and Homenda, W. Operations on balanced fuzzy sets. In *2023 IEEE International Conference on Fuzzy Systems (FUZZ)*, pages 1–6. IEEE, 2023.

[6] Matusiewicz, Z. and Homenda, W. Balanced relations equations. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2022.

[7] Matusiewicz, Z. Balanced and intuitionistic fuzzy systems of equations. In *International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets*, pages 22–31. Springer, 2022.

[8] Pawlak, Z. *Rough sets: Theoretical aspects of reasoning about data*, volume 9. Springer Science & Business Media, 2012.

[9] Risk factors, British Heart Foundation. URL `https://www.bhf.org.uk/informationsupport/risk-factors`.

Rough Set Approach to Scalable Similarity Measures – The Case Study of Interactive Labeling

Dominik Ślęzak[1234][0000−0003−2453−4974]

[1]*QED Software, Poland*
[2]*Institute of Informatics, University of Warsaw, Poland*
[3]*OnstageAI, USA*
[4]*DeepSeas, USA*

Abstract. *We discuss how the ensembles of rough-set-based reducts can support the interactive data labeling processes. We suggest to use such ensembles not only as decision models, but also to define a specific notion of similarity, which in turn can help us to select data cases to be labeled.*
Keywords: *interactive data labeling, uncertainty, similarity, rough sets*

1. Introduction

Ability to handle similarities is crucial in many machine learning scenarios. For instance, in interactive labeling [1] (powered by active learning [2]), it is useful to focus on the not-yet-labeled data cases that are (A) not similar to the already-labeled ones or (B) similar to the labeled cases with respect to which a current model is uncertain. As another example, when one wants to diagnose a cause of mistake of a machine learning model that happened for a given current case, it is a good idea to analyze similar cases in the model's performance history [3].

We consider such scenarios from the viewpoint of making the underlying similarity computations scalable and insightful. We discuss the idea of expressing similarities, which is based on the ensembles of decision reducts – one of the fundamental notions in the theory of rough sets [4]. We discuss some variations of decision reducts (such as approximate reducts and bireducts [5]) that are utilized

DOI:10.17388/WUT.2024.0002.MiNI

to learn simple decision models from the data. On top of that – which is of our special interest in this paper – they form data partitions based on easily interpretable attribute subsets. Accordingly – somewhat analogously to the LSH methods [6] – their ensembles can form the data-partition-based similarities.

2. Active Learning and Interactive Labeling

Active learning is a powerful machine learning methodology, whereby decision models are learnt incrementally, on incrementally increasing data sets [2]. One of the paradigms of active learning is that it is not always the best idea to rely on the whole available data for training. Instead, the active learning algorithms should be allowed to decide which data cases are most useful for them and query for their labels. Naturally, the idea of active learning goes well along with these of real-world applications, whereas the resources of labeled data are limited and one needs to apply some mechanisms for choosing the data to be labeled anyway [1]. To give a few examples, one can consider a task of learning from subject matter experts (SMEs) which alerts correspond to truly dangerous network events[1] (see also [7]), or a task of annotating which musicians should be displayed in particular moments during a video transmission of a musical concert[2] (see also [8]).

Let us discuss some triggers that can imply a choice of a given data case while deciding how to optimally facilitate the time of SMEs:

- A data case may be worth showing to an SME if it is not similar to any data cases that were considered in the learning process up to now;
- A data case may be worth showing to an SME if it is similar to a group of other cases labeled by that SME up to now, but the SME seemed to be quite uncertain about those cases (see also [9]);
- A data case may be worth showing to an SME if it is similar to a group other cases that were analyzed by the previous versions of a decision model that we attempt to learn, but the model seemed to be quite uncertain about those cases (see also [10]).

Certainly, these are only a few examples of situations that should be taken into account while designing an automatized system for interactive data labeling. As

[1] https://www.deepseas.com/
[2] https://www.onstageai.com/

we can see, these examples require a good handling of the notion of uncertainty and even more importantly – the notion of similarity. In particular, the measures of uncertainty are usually based on the analysis of posterior distributions or densities associated to data cases [11,12]. However, such distributions are often estimated from the labels (being a result of the SMEs' annotations or the decision model inferences) observed on the other most similar data cases.

3. Rough Sets and Reduct-based Similarities

Rough set methods often refer to decision reducts that are irreducible subsets of conditional attributes (available features, variables) that deliver the same amount of information about decision attributes (target/dependent variables) as the whole sets of conditional attributes do [4]. Among various extensions of decision reducts, one can point at approximate reducts (that can lose a controlled amount of information) and bireducts (that report non-extendable subsets of data cases for which the given irreducible subsets of conditional attributes yield valid decision rules) [5]. In applications, it is particularly useful to rely on the ensembles of possibly smallest approximate reducts (or bireducts) that are based on possibly most diverse conditional attributes. Compared to the other types of ensembles [13], the ensembles of (bi)reducts can provide more interpretable decision models [14] and, e.g., more insightful rankings of conditional attributes [15].

In [3,16], it was noted that the ensembles of diverse decision reducts may be utilized also for approximating a work of the monitored machine learning models (or solutions submitted to the data science competitions[3]). The idea is to learn an ensemble that behaves in a similar way as the original model and then use it for explaining why that model might make mistakes. Again, one possible reason for making a mistake on a given data case is that the model did not examine similar cases before, while another reason may be related to the model's uncertainty. As each single reduct yields a collection of the if-then decision rules associated with combinations of values over conditional attributes belonging to the reduct, we can look for the historical cases that matched the same reduct-based rules as the case diagnosed currently. The more reducts yield common rules for two data cases, the closer these cases are to each other. This kind of reduct-based similarity makes it possible to compose the neighborhoods of data cases and conduct explainable diagnostics based on the sizes and contents of these neighborhoods.

[3] https://knowledgepit.ai/

The thesis of this paper is that we can follow the analogous approach in active learning and interactive labeling. Namely, if we use the decision-reduct-ensemble method of learning a decision model within the active learning loop, we can apply this ensemble itself to define a similarity over the space of available data cases. Then, we can rely on such designed similarity like it was explained in Section 2. The only remaining aspect is to be able to compute ensembles of decision reducts sufficiently fast over large amounts of the data and in the next step, utilize them to quickly derive similarity coefficients. Luckily, there exist software libraries for fast reduct calculations (see e.g. [15]) and when it comes to the similarity derivations, the analogies with the LSH framework can be helpful [6].

4. Conclusions and Future Research Directions

We discussed how to apply the algorithms that learn the ensembles of decision reducts from the data to conduct the active learning and interactive data labeling processes. We followed the idea of using such ensembles not only as the decision models, but also as the basis for introducing the notion of similarity, which can help us to operate with efficient and scalable uncertainty modeling.

We intend to introduce such rough-set-based mechanisms into our data science platform called BlueQuail[4]. However, we first need to conduct more research on the uncertainty measures. Currently, our methods for measuring uncertainty are different when it comes to the machine learning model diagnostics [3] and interactive labeling [9]. If we want BlueQuail to support both such functionalities, we need a standardized measure, preferably based on the considered ensembles of decision reducts that can be handled by the BlueQuail's backend layer.

Another research direction refers to the paradigm of query-by-example (QBE [17]) which can be based on reduct-based similarities too. Herein, interpretability of the ensembles of decision reducts can be particularly important, as the QBE-users may require explanations why some data cases were assessed as similar or dissimilar. Such explanations can be important also in the case of interactive data labeling. For example, an SME may ask why she or he receives a given set of data cases to be labeled while the underlying system may need to explain to her or him that these cases have been assessed as the least similar to the previous ones.

Finally, it is worth emphasizing that interactions with SMEs may be valuable for the machine learning processes not only when it comes to data labeling. In the

[4]https://www.bluequail.ai/

previous sections, we pointed out that the ensembles of decision reducts are easy to interpret because they straightforwardly correspond to the subsets of attributes used in the decision model's inference. However, interpretable reducts should consist of interpretable attributes and therefore, it may be important to attempt to get SMEs involved also at the stage of interactive attribute selection [18].

Acknowledgment

This research was co-funded by Smart Growth Operational Programme 2014-2020, financed by European Regional Development Fund, in frame of project POIR.01.01.01-00-0213/19, operated by National Centre for Research and Development in Poland.

References

[1] Assenmacher, I., Hentschel, B., Ni, C., Kuhlen, T.W., Bischof, C.H. Interactive Data Annotation in Virtual Environments. *12th Eurographics Symposium on Virtual Environments – EGVE 2006*, pages 119–126. Eurographics Association, 2006. doi:10.2312/EGVE/EGVE06/119-126.

[2] Settles, B. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2012. ISBN 978-3-031-00432-2.

[3] Janusz, A., Zalewska, A., Wawrowski, Ł., Biczyk, P., Ludziejewski, J., Sikora, M., Ślęzak, D. BrightBox – A Rough Set based Technology for Diagnosing Mistakes of Machine Learning Models. *Applied Soft Computing*, 141:110285, 2023. doi:10.1016/J.ASOC.2023.110285.

[4] Skowron, A., Ślęzak, D. Rough Sets Turn 40: From Information Systems to Intelligent Systems. *17th Conference on Computer Science and Intelligence Systems – FedCSIS 2022*, pages 23–34. IEEE, 2022. doi:10.15439/2022F310.

[5] Stawicki, S., Ślęzak, D., Janusz, A., Widz, S. Decision Bireducts and Decision Reducts – A Comparison. *International Journal of Approximate Reasoning*, 84:75–109, 2017. doi:10.1016/J.IJAR.2017.02.007.

[6] Ullman, J. Big-Data Algorithms That Are Not Machine Learning. In *IEEE International Conference on Big Data – IEEE Big Data 2023*, pages 5. IEEE, 2023. doi:10.1109/BIGDATA59044.2023.10386233.

[7] Janusz, A., Kałuża, D., Chądzyńska-Krasowska, A., Konarski, B., Holland, J., Ślęzak, D. IEEE BigData 2019 Cup: Suspicious Network Event Recognition. In *IEEE International Conference on Big Data – IEEE Big Data 2019*, pages 5881–5887. IEEE, 2019. doi:10.1109/BIGDATA47090.2019.9005668.

[8] Fiebig, J., Borowski, A. Ślęzak, D., Bukała, A., Kowaleczko, P., Kwiatkowska, Z., Ludziejewski, J., Zalewska, A. System and Method for Camera Handling in Live Environments. *US Patent Application*, 20240080566. USPTO, 2024.

[9] Kałuża, D., Janusz, A., Ślęzak, D. Robust Assignment of Labels for Active Learning with Sparse and Noisy Annotations. In *26th European Conference on Artificial Intelligence – ECAI 2023*, pages 1207–1214. IOS Press, 2023. doi:10.3233/FAIA230397.

[10] Nguyen, V.-L., Shaker, M.H., Hüllermeier, E. How to Measure Uncertainty in Uncertainty Sampling for Active Learning. *Machine Learning*, 111(1):89–122, 2022. doi:10.1007/S10994-021-06003-9.

[11] Kałuża, D., Janusz, A., Ślęzak, D. On Several New Dempster-Shafer-Inspired Uncertainty Measures Applicable for Active Learning. *International Joint Conference on Rough Sets – IJCRS 2023*, pages 479–494. Springer, 2023. doi:10.1007/978-3-031-50959-9_33.

[12] Wu, J., Chen, J., Huang, D. Entropy-based Active Learning for Object Detection with Progressive Diversity Constraint. *IEEE/CVF Conference on Computer Vision and Pattern Recognition – CVPR 2022*, pages 9397–9406. IEEE, 2022. doi:10.1109/CVPR52688.2022.00918.

[13] Konstantinov, A.V., Utkin, L.V. Interpretable Machine Learning with an Ensemble of Gradient Boosting Machines. *Knowledge-Based Systems*, 222:106993, 2021. doi:10.1016/J.KNOSYS.2021.106993.

[14] Ślęzak, D., Grzegorowski, M., Janusz, A., Kozielski, M., Nguyen, S.H., Sikora, M., Stawicki, S., Wróbel, Ł. A Framework for Learning and Embedding Multi-Sensor Forecasting Models into a Decision Support System: A

Case Study of Methane Concentration in Coal Mines. *Information Sciences*, 451-452:112–133, 2018. doi:10.1016/J.INS.2018.04.026.

[15] Janusz, A., Ślęzak, D., Stawicki, S., Stencel, K. A Practical Study of Methods for Deriving Insightful Attribute Importance Rankings Using Decision Bireducts. *Information Sciences*, 645:119354, 2023. doi:10.1016/J.INS.2023.119354.

[16] Janusz, A., Ślęzak, D. KnowledgePit Meets BrightBox: A Step Toward Insightful Investigation of the Results of Data Science Competitions. *17th Conference on Computer Science and Intelligence Systems – FedCSIS 2022*, pages 393–398. IEEE, 2022. doi:10.15439/2022F309.

[17] Tyagi, V. Content-Based Image Retrieval – Ideas, Influences, and Current Trends. Springer, 2017. doi:10.1007/978-981-10-6759-4.

[18] Ślęzak, D., Grzegorowski, M., Janusz, A., Stawicki, S. Toward Interactive Attribute Selection with Infolattices – A Position Paper. *International Joint Conference on Rough Sets – IJCRS 2017*, vol. 2, pages 526–539. Springer, 2017. doi:10.1007/978-3-319-60840-2_38.

Chapter 8

Robotics and Autonomous Systems

Track Chairs:

- prof. Piotr Skrzypczyński - Poznan University of Technology
- prof. Piotr Lipiński - Lodz University of Technology
- prof. Cezary Zieliński - Warsaw University of Technology

Hierarchical TMP: combining HTN and geometric planning

Dawid Seredyński[0000−0003−2528−6335]

Warsaw University of Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19, 00-665 Warsaw, Poland
dawid.seredynski@pw.edu.pl

Abstract. *The paper presents a Task and Motion Planning (TMP) technique based on Hierarchical Task Network (HTN) and conditional generators. The HTN notation is based on literature and it is extended with a world model that fills the "semantic gap" between a symbolic level of task planning and a geometric level of motion planning. The presented planer has been validated on a challenging planning domain of service robotics in kitchen environment. Presented experiments show that generation of parameters of tasks (geometric planning) is the most time-consuming operation and feedback information from generators improve performance of the planner.*
Keywords: *task and motion planning, robotics, artificial intelligence, conditional generators, streams, hierarchical planning, world model*

1. Introduction

Task and motion planning (TMP) is indispensable in robotics, as modern robots perform complex tasks in unstructured environment. TMP combines discrete task planning, motion planning in continuous space, and optimization [1]. There are basically three main problems with TMP that make it challenging: (i) huge space of states, actions and their parameters, (ii) high complexity of tasks that comes from the diversity of environments and situations, and (iii) a "semantic gap": a plan that is valid at a symbolic level (discrete tasks), may be geometrically infeasible (continuous motion).

The presented approach (presented in detail in the PHD thesis [2]) utilizes a hierarchical approach to task planning: hierarchical task networks (HTN) [3, 4]. It

DOI:10.17388/WUT.2024.0002.MiNI

is a domain configurable planning technique that is built of universal algorithms suited to any problems that can be specified using a specific input language, and it allows to encode large portion of knowledge, "recipes" for solving tasks, denoted as a "planning domain", specified in a formal language, e.g. HDDL [4]. The presented work shows that this knowledge may include routines of geometric planning, seen here as not only motion planning, but also grasp selection, selection of direction of approach to an object, inverse kinematics, determination of a pose where an object should be placed, etc. Joining task planning and geometric planning requires spanning the semantic gap between these two levels. Task planning is realized with HTN, while the geometric planning uses a concept of conditional generators, based on PDDLStream [5]. A world model, added to the HTN formalism, fills the semantic gap, and it encapsulates black boxes: conditional generators (streams) that generate parameters for tasks, and predicate evaluation function for "extended" predicates, that are related to geometric features of environment.

The general idea of forward, state-based progression search HTN planning is similar to the one used by the SHOP2 planner [3]. Utilization of generators, equipped with a declarative part is adopted from PDDLStream [5], and external procedures for evaluation of truth value of predicates is based on semantic attachments presented in [6]. HTN notation is similar as in many other sources on this subject [4, 6]. The input language is similar to HDDL (an extension of PDDL for hierarchical planning problems [4]), with some elements adopted from PDDLStream [5]. The original work on PDDLStream [5] presents a planner that uses streams to ground free parameters of operators of classical planning. The concept of streams is adopted in the presented work to HTN. A closely related work, to the presented approach, describes semantic attachments for HTN planning (HyperTensioN planner [6]). It is focused on grounding a HTN problem using semantic attachments that are implemented as coroutines, that is functions that can suspend their execution, saving their internal state and yielding some value, and then resume execution. That is a similar concept to PDDLStream, except that the latter is equipped with declarative components that describe required properties of inputs and specifies properties of outputs.

The presented work extends state of the art in the following. It joins PDDLStream with HTN and semantic attachments, by integrating streams into decomposition methods, and utilizes a world model with predicate evaluation function and black-box generators for streams. This enables the generation of parameters for both complex and primitive tasks, at all levels of task hierarchy, thus parameters can represent mathematical objects at various levels of abstraction. Streams

may generate parameters for actions, that is, tasks at the lowest ontological level, utilizing as an input previously generated parameters of complex tasks, generated by other streams, at higher ontological levels.

2. Task and motion planning formalism

A problem of task and motion planning can be simply expressed as: for a given task, determine sequence of movements $a_1, a_2, ..., a_n$ that starting from the initial state s, accomplishes the task. The sequence $a_1, a_2, ..., a_n$ is called a plan. Formally, a **planning problem** is a triple $\mathcal{P} = (\mathcal{D}, s, tn)$, where $\mathcal{D}$ is a planning domain (1), s is a world state, and tn is a task network that specifies tasks to be done. A **world state** is a set of facts (ground, common predicates, $s \subset PR_G^{com}$) that describe state of environment and robot at some time instant. A **task network** is a partially ordered set of tasks that represent tasks to be done and their order. A **planning domain** organizes knowledge about solving planning problems and it is a quadruple (1)

$$\mathcal{D} = (\mathcal{L}(S), OP, M, \mathcal{W}), \tag{1}$$

where $\mathcal{L}(S)$ is language over alphabet $S = S_V \cup S_C \cup S_{PR} \cup \{\wedge, \neg\}$ that is build of logical connectors $\wedge, \neg$, and symbols of: variables S_V, constants S_C and predicates S_{PR}. A set of operators OP relate primitive tasks (actions) to their effects, and a set of decomposition methods M define hierarchy of tasks. A **world model** is a triple $\mathcal{W} = (C, \mathcal{G}, O)$, where function $C : ST \times PR_G^{ext} \rightarrow \{\texttt{True}, \texttt{False}\}$ calculates value of an extended predicate $\in PR_G^{ext}$ in a given state $s \in ST$, $\mathcal{G}$ is a procedural part of streams, and O is a set of mathematical objects related to symbols of constants $x \in S_C$. The world model is a black box with a specified interface, that fills the "semantic gap" between the symbolic level ($x \in S_C$) and the geometric level ($\text{obj}(x) \in O$). The world model in the HTN formalism is a novel element.

An **extended predicate** is a predicate evaluated by a black-box procedure C. A **common predicate** is a predicate that is not an extended predicate. A set of all ground, common predicates is PR_G^{com}. An extended predicate is not a part of world state s, however, it may be evaluated by C for a given s, e.g. `IsSafe(rob, tr1)` – "is a given trajectory `tr1` safe for robot `rob`?" requires non-trivial calculations, e.g. collision detection between robot and its surroundings for the whole trajectory.

A **primitive task (action)** may be directly transformed into a behavior of a robot and it is specified by an **operator** $(\text{task}(o), \text{pre}(o), \text{eff}(o))$, where $\text{task}(o)$ specifies a task class and its parameters, $\text{pre}(o)$ is a precondition, and $\text{eff}(o)$ spec-

ifies effects. An operator can be interpreted as a relation between two states. Negated predicates in eff(o) are called negative effects (denoted by $\text{eff}^-(o)$), and the others are called positive effects (denoted by $\text{eff}^+(o)$). When an operator is applied in some state s_1, a resultant state s_2 is created with deleted facts from $\text{eff}^-(o)$ and added facts from $\text{eff}^+(o)$. This can be interpreted as a state transition from s_1 to s_2.

A **complex task** is subject to decomposition into a task network. A **decomposition method** (2) is a tuple:

$$(\text{name}(m), \text{task}(m), \text{pre}(m), \text{net}(m), \text{str}(m)), \tag{2}$$

that specifies how to decompose a given task task(m), under circumstances specified as a precondition pre(m), into a task network net(m), with task parameters generated by streams given in str(m). An unique name of a method and its parameters are defined in name(m). A task network that is a result of decomposition of a task may have more parameters than the task. To generate values for the new parameters, a decomposition method contains a sequence of streams $\text{str}(m) = \langle \sigma_1, ..., \sigma_n \rangle$. A **stream** σ is a tuple (3):

$$(\text{inp}(\sigma), \text{domain}(\sigma), \text{out}(\sigma), \text{cert}(\sigma), \text{err}(\sigma)), \tag{3}$$

where inp(σ) is a syntactic expression $f\left(x_1, ..., x_{|f|-1}\right)$ where f is a function symbol of state-dependent conditional generator and $x_1, ..., x_{|f|-1} \in S_V$ are symbols of parameters (inputs). A **state-dependent conditional generator** $f : ST \times C_{|f|-1} \to G_n$ is a function that relates states and object tuples to generators of object n-tuples. domain(σ) is a formula that specifies properties that inputs $x_1, ..., x_{|f|-1}$ must satisfy, out(σ) is a n-tuple of symbols of variables $y_1, ..., y_n \in S_V$ that represent elements of output tuples, cert(σ) is a set of predicates that specifies (certifies) properties that inputs $x_1, ..., x_{|f|-1}$ and outputs $y_1, ..., y_n$ must satisfy in conjunction. In the presented work, an error reporting function is added to stream: err(σ) is either $\emptyset$ or an $(|f|-1)$-tuple $\left(e_1, ..., e_{|f|-1}\right)$ of predicates $e_i : ST \times C_{|f|-1} \to \{\texttt{True}, \texttt{False}\}$ that indicate whether a given i-th input parameter is the cause of failure; if err(σ) is $\emptyset$, then no information about reason of failure is available.

A **planning problem** $\mathcal{P} = (\mathcal{D}, s, tn)$ **is solvable** if tn is empty or there is a planning problem $\mathcal{P}'$, such that $\mathcal{P} \to \mathcal{P}'$ and $\mathcal{P}'$ is solvable. There are three kinds of $\mathcal{P} \to \mathcal{P}'$: execution of action, decomposition of complex task, and grounding of parameters of task network. The search algorithm is a state-based progression search, where branching points include choice of a method of decomposition and

choice of grounding of parameters. The search algorithm builds a tree of planning problems and in each step it selects and progresses a planning problem from a fringe, and adds the resultant problem to the tree.

3. Experiments

The presented task and motion planning formalism has been validated on a two-arm service robot WUT Velma [7], and a challenging planning domain of service robotics, in kitchen environment. Experiments were conducted in environments with movable objects, and tight spaces, such as interior of a cabinet. The planning domain contained 19 complex tasks, 7 operators, 33 decomposition methods and 37 streams. Tests were performed on a number of test cases that varied in complexity. Some of the generated plans were also executed on a real H/W platform (fig. 1) Test cases are exemplary problems generated for a number of scenarios

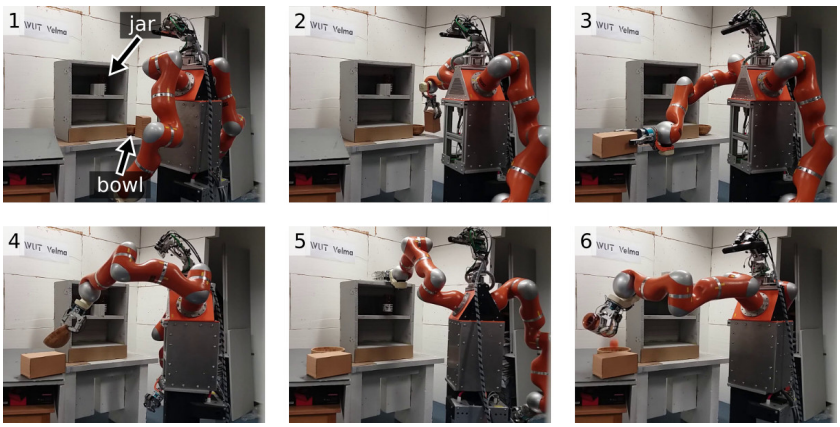


Figure 1. Execution of a complex task: pour from a jar into a bowl (1). Both objects are behind other objects, thus the robot has to rearrange the environment (2, 3, 5), before it can grasp the bowl (4) and the jar (6)

of manipulation in the kitchen environment containing: tables, cabinets and movable objects, such as boxes, jars and bowls. Those examples were selected so that the performance of the presented planner and algorithms could be assessed. They utilize all of the tasks present in the exemplary planning domain. For all experi-

ments, it was assumed that state of the environment is fully known, and the system used the look-and-go principle, that is, planning was performed once, and then the fully generated plan was executed by the robot. Execution of the plan was done by moving the robot along the planned paths, as specified by the sequence of actions in the plan. For the easiest problems, the generated plans consisted of 5 actions, and the hardest contained 66 actions. In the hardest cases the robot had to cleanup some parts of its environment in order to perform the requested task (as shown in fig. 1).

Three algorithms for the selection of planning problems in the search tree (elements in the fringe) were compared. Execution time of planning is nondeterministic, because the planning algorithms use a non-deterministic algorithms for selection of nodes in the search tree, and some generators of samples (streams) use non-deterministic algorithms, e.g. RRT-Connect for path planning. Thus, mean time from 5 runs is presented. Due to large computation time for some of the algorithms and for some test cases, the time limit was set to 700 s.

Performance of the algorithms is presented in fig. 2. The first algorithm, denoted as `selectDfs`, selects a node and progresses the associated planning problem using depth first search. It quickly deepens the search tree. However it may get stuck in exploring the deepest nodes, and infinitely sampling parameters for problems that are not possible to solve, due to wrong parameters selected at earlier stages of search. It is the quickest for some test cases (P, M, Q, S), but sometimes it fails to find a solution (B, R, T). The reason is that, it got stuck in deep nodes of the search tree, and it could not backtrack to a node, where infeasible parameters were selected at earlier stages of planning. The second algorithm `selectGreedyRnd` combines greedy search and selection of random nodes from the fringe. It selects the last added node, if the last progression succeeded, or otherwise, it selects a random node using a non uniform probability distribution that favors deeper nodes. It behaves poorly for the largest problems (Q, R, S, O, U), or even it cannot find a solution within time limit (test case T), because if greedy selection fails, it blindly selects a node, preferring deep nodes in the search tree. The third algorithm `selectGreedyRndB` uses information about the reason of failure of a conditional generator given by err(σ). It backtracks to nodes in the search tree that produced parameters that might have caused the failure of CG in the last visited node. If the reason of failure is unknown, `selectGreedyRnd` is used. The algorithm `selectGreedyRndB` shows better performance than `selectGreedyRnd` for the largest problems, because it uses additional information about possible reason of failure of a stream.

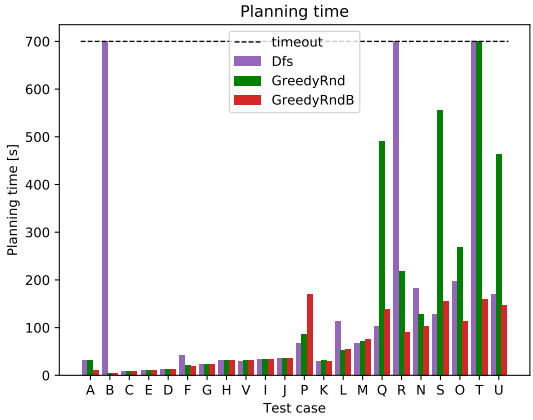


Figure 2. Planning times for all test cases and three algorithms for node selection

Experiments show that the planner can solve complex planning problems in environments with movable objects and obstacles. The concept of failure detection for streams (err(σ) in eq. (3)) improves performance of the TMP, as it guides the search using information on wrongly selected parameters (the algorithm `selectGreedyRndB`). The experiments also have shown that 97% of the planning time is consumed on geometric planning (realized by streams), e.g. trajectory planning (72% of the whole planning time), determining where to place an object (17%), grasp planning (4%), determining direction of approach (3%), inverse kinematics (1%).

4. Conclusions

The presented planning technique is suitable for task and motion planning for a service robot. The experiments have shown that most of the time is consumed on generation of task parameters and planning performance may be improved by using a feedback information from black-box conditional generators in streams. A great challenge in TMP is the semantic gap between the geometric level and the symbolic level. Thus, understanding of relations between those two levels is crucial in TMP. The future work may include utilization of Large Language Models

(LLM) to reason about causes of failures at the geometric level and to utilize this information at the symbolic level. Another interesting topic is interleaved planning and execution that allows to generate incomplete plans, where a few first tasks are defined at most detailed level (as actions), and the rest of the plan is roughly specified. This allows to model uncertainty and incomplete knowledge and it enables discovering knowledge during task execution.

References

[1] Garrett, C. R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L. P., and Lozano-Pérez, T. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[2] Seredyński, D. *Planning and Realization of Service Robot Manipulation Tasks at Multiple Levels of Abstraction*. Ph.D. thesis, Warsaw University of Technology, 2023.

[3] Nau, D. S., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., and Yaman, F. Shop2: An htn planning system. *Journal of artificial intelligence research*, 20:379–404, 2003.

[4] Höller, D., Behnke, G., Bercher, P., Biundo, S., Fiorino, H., Pellier, D., and Alford, R. Hddl: An extension to pddl for expressing hierarchical planning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9883–9891. 2020.

[5] Garrett, C. R., Lozano-Pérez, T., and Kaelbling, L. P. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448. 2020.

[6] Magnaguagno, M. C. and Meneguzzi, F. Semantic attachments for htn planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9933–9940. 2020.

[7] Winiarski, T., Jarocki, S., and Seredyński, D. Grasped object weight compensation in reference to impedance controlled robots. *Energies*, 14(20):6693, 2021.

Towards Efficient Generation of Data Using Mixed Domains for Segmentation of 3D LiDAR Point Clouds

Konrad Cop[1,2][0000−0001−8159−9307], **Bartosz Sułek** [2][0009−0006−8005−0914] **and Tomasz Trzciński** [1][0000−0002−1486−8906]

[1]*Warsaw University of Technology*
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
[2]*United Robots Sp. z o.o.*
Świeradowska 47, 02-622 Warsaw, Poland
cop.konrad@gmail.com

Abstract. *Semantic segmentation is important for robots navigating with 3D LiDARs but the generation of training datasets requires tedious manual effort. In this paper, we introduce a method to efficiently generate large datasets by locally augmenting available empty scenes with navigation-relevant obstacles. Our approach uses a combination of real and synthetic data and leads to satisfactory segmentation performance with no manual annotation or no objects of interest whatsoever. We validate the proposed method on the underground parking scans and compare it with available open-source datasets. Our solution is suitable for direct application in robotic systems and opens new research areas.*
Keywords: *deep learning for visual perception, semantic segmentation, 3D LiDAR, robotic perception*

1. Introduction

In robotic 3D perception, semantic segmentation of point clouds is crucial to make relevant navigation decisions. Usually, point cloud segmentation is defined as classifying each point into one of multiple distinct categories [1]. In this paper, we reformulate the problem to make it applicable in a practical robotic context by

DOI:10.17388/WUT.2024.0002.MiNI

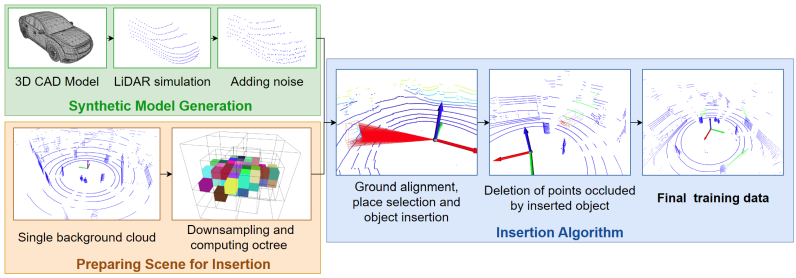


Figure 1. Process of training data generation through the integration of synthetic 3D objects into recorded background point clouds

focusing only on the classification of specific dynamic obstacles and treating the remaining points as not crucial. This approach originates from the challenges of a mobile robot operation, where the efficiency deteriorates the most when the robot must diverge from the intended path due to unexpected obstacles [2, 3]. In such a case, classifying dynamic obstacles can help the robot make correct navigation decisions while distinguishing the objects that remain constantly static e.g. walls, pillars, furniture, etc. is not equally relevant.

Achieving successful segmentation performance requires vast training datasets, which can be generated by manual annotation of data samples. This is especially time-consuming for 3D LiDAR data and becomes prohibitively tedious due to the variety of possible environments [4]. A directly correlated issue might be the lack of instances of the object of interest in the deployment area to sufficiently train the model. There exists another way of datasets generation that utilizes simulation frameworks in which the complete appearance of any environment can be created and the sensor returns simulated [5, 6]. Although it is undoubtedly faster than real dataset annotation, the synthetic approach suffers from two main drawbacks. The proper "virtual world" must be laboriously generated using 3D CAD models of different objects and the synthetic representation experiences a reality gap originating from the inability to fully model complex noise characteristics of sensor perception of real objects.

To address the limitations of these two approaches we propose a combined method that utilizes data from the real domain, namely 3D LiDAR scans, which are augmented with synthetically generated instances of the objects of interest. In this way, synthetic objects are directly annotated while the background scans

are classified as “unknown" and the dataset requires no additional manual work. Our system assumes that the background scans are available, which is usually the case when a robot is deployed in a new environment and the navigation map is created. As a result, a meaningful training dataset can be created from scratch, provided that proper synthetic models of objects are available. A similar concept to ours was proposed by Fang et al. [7], with the difference that the authors select instances' insertion places using learned distributions, which ultimately requires a labeled dataset. Our approach relies solely on statistical and geometrical methods for insertion and no additional dataset is necessary.

To verify the approach we collected data in an underground parking environment and augmented it with synthetically generated car instances. Subsequently, we trained the segmentation network with these data to validate the performance. For a broader context, we also compared the results with manually annotated ground truth and open-source datasets of similar characteristics. The following sections explain the method in detail and describe experimental verification.

2. Data Generation

Our approach consists of the insertion of synthetic object instances within real “background clouds”. In this paper, we specifically focus on the parking scenario. The synthetic cars were generated at different distances and angles using the Gazebo simulation framework, simulating VLP-16 LiDAR readings on 3D CAD models from ShapeNet [8], while the background clouds were recorded during the actual operation of the robot. From ShapeNet dataset only *Car* models were used, specifically the classes: *Car, Sedan, Coupe,* and *Hatchback*. These were manually checked for appropriate instances since ShapeNet contains many corrupted models, which resulted in 475 proper instances. One should note, that the system operates on single LiDARs scans, obtained from the sensor with a working frequency 20 Hz, therefore we assume that in each scan, the object instances remain static and no additional motion compensation is needed. To bridge the reality gap that arises from challenges like an accurate simulation of reflections, the presence of windows, or a high variety of cars design, the process as shown in Figure 1, involves several steps to ensure training scenarios are realistic. Initially, using the computed octree structure of a background cloud, we perform ground alignment and transformations allowing for the precise placement of car models at various angular positions to produce several candidate locations.

Training dataset	Acc	IoU
UR Dataset (Ground Truth)	95.2	91.0
S-KITTI	79.5	52.6
Pandaset	75.2	70.7
UR background + Pandaset + S-KITTI	56.2	45.8
UR background, Shapenet cars (WoN)	57.8	55.0
UR background, Shapenet cars (WN)	<u>73.1</u>	<u>64.3</u>
UR background, Shapenet cars (WN) + Pandaset + S-KITTI	81.3	72.9

Table 1. Results of the verification on test subset of UR dataset, using two metrics Accuracy (Acc) and Intersection over Union (IoU). We use UR background (i.e. real scans without cars) in combination with synthetic instances as well as SemanticKITTI and Pandaset datasets. Synthetic Shapenet clusters are either Without Noise (WoN) or With Noise (WN) before insertion. We apply solely the geometrical and statistical noise as described in the Section 2. The last row corresponds to the combined dataset. The results of our method are underlined.

Following this, every candidate undergoes an assessment for potential collisions with the background and surrounding objects. The insertion algorithm verifies that the chosen location isn't obstructed by existing points, ensuring clear spaces prior to placement. Then, points occluded by newly inserted objects are removed to simulate real LiDAR performance. The direct consequence of our method is the equal likelihood of the place where the instance will be inserted provided that this area is not occupied by any other object. Finally, to mimic real sensor noise, we add per-point dropout and noise sampled from a modeled normal distribution, applied by XYZ-shift augmentations on both the instances and the whole map [9]. It's worth mentioning that the insertion algorithm will also work with real instances e.g. cars extracted from open-source datasets.

3. Experiments

To verify our approach we created a ground truth dataset, which we refer to as the *UR dataset*, using real 3D LiDAR scans of underground parking and annotating all occurrences of cars. Overall, it consists of 6 670 scans containing 46 593 car instances, divided into train, validation, and test subsets in the proportion 70/15/15%. The manual annotation took 32 hours. From other recorded

scans we extracted 6 740 background scans without cars, which we used for data generation following the method described in Section 2. Synthetic instances were generated in two ways namely with and without additional noise. We also used two open-source datasets i.e. SematicKITTI [10] and Pandaset [11], to examine the potential of Transfer Learning from different domains. Finally, we merged multiple datasets in different combinations. On all sets, we trained the SphereFormer [12] segmentation network and ran it on the test subset of the UR dataset. All discussed experiments were trained from scratch for approximately 20 to 30 epochs, using early stopping to prevent model overfitting. Following the metrics suggested by the authors, we used Accuracy and IoU for numerical comparison. Table 1 summarises the results. The best performance is achieved with training on the dataset from the same domain as the test set (UR dataset). Mediocre results of open-source datasets we associate with different characteristics of the sensors as our LiDAR has 16 beams while the others 64, and with the difference of the background since both datasets cover urban (outdoor) environments contrary to our, solely indoor spaces. As for the mixed-domain dataset, the naive approach of inserting synthetic, instances of cars without noise is far from perfect, however adding multiple noise modelling techniques significantly improves the result. Even though the reality gap is not fully addressed it proves the proposed approach is valid, nevertheless, a complete noise model of the sensors is required to fully imitate the actual dataset. An example of segmentation is shown in Figure 3. Finally, the tests of multiple datasets combined show better results than each treated separately which suggests good generalization capabilities.

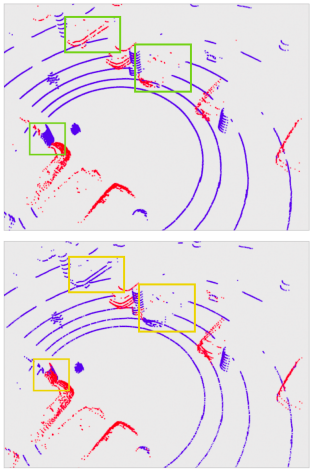


Figure 2. Example of a segmented point cloud. Top: UR dataset ground truth data. Bottom: prediction result of the model trained with real background scans and synthetic instances with noise. It is visible that the system fails with small and spatially shrunk clusters.

4. Conclusions

Our method results in good performance of segmentation training and allows to quickly generate training datasets with no manual effort. The mixed domains approach reaches a similar level as real datasets from alike but not identical distributions. We also showed that combining different sources of 3D LiDAR data with similar characteristics creates an efficient solution for training dataset generation even in the complete absence of annotated data samples from the actual environment or even occurrences of objects of interest at the time of data recording. With the given dataset we managed to reduce annotation time from 32 hours to virtually no time, at the cost of accuracy. Thus our approach is applicable for a quick setup of a segmentation system in a robot. The results highlight open research avenues, namely generalization to other types of obstacles and environments and realistic sensor noise modeling.

Acknowledgment

The authors would like to thank Jakub Frąszczak for his support with the implementation of the car insertion algorithm.

References

[1] Zhang, J., Zhao, X., Chen, Z., and Lu, Z. A review of deep learning-based semantic segmentation for point cloud. *IEEE access*, 7:179118–179133, 2019.

[2] Kunchev, V., Jain, L., Ivancevic, V., and Finn, A. Path planning and obstacle avoidance for autonomous mobile robots: A review. In *Knowledge-Based Intelligent Information and Engineering Systems: 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006. Proceedings, Part II 10*, pages 537–544. Springer, 2006.

[3] Badrloo, S., Varshosaz, M., Pirasteh, S., and Li, J. Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review. *Remote Sensing*, 14(15):3824, 2022.

[4] Ibrahim, M., Akhtar, N., Wise, M., and Mian, A. Annotation tool and urban dataset for 3d point cloud semantic segmentation. *IEEE Access*, 9:35984–35996, 2021.

[5] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[6] Xiao, A., Huang, J., Guan, D., Zhan, F., and Lu, S. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2795–2803. 2022.

[7] Fang, J., Zhou, D., Yan, F., Zhao, T., Zhang, F., Ma, Y., Wang, L., and Yang, R. Augmented lidar simulator for autonomous driving. *IEEE Robotics and Automation Letters*, 5(2):1931–1938, 2020.

[8] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] Alonso, I., Riazuelo, L., Montesano, L., and Murillo, A. C. Domain adaptation in lidar semantic segmentation by aligning class distributions, 2021.

[10] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307. 2019.

[11] Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021.

[12] Lai, X., Chen, Y., Lu, F., Liu, J., and Jia, J. Spherical transformer for lidar-based 3d recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17545–17555. 2023.

Rico: extended TIAGo robot towards up-to-date social and assistive robot usage scenarios

Tomasz Winiarski[0000−0002−9316−3284],
Wojciech Dudek[0000−0001−5326−1034],
Daniel Giełdowski[0000−0002−4348−2981]

Warsaw University of Technology
Faculty of Electronics and Information Technology
Institute of Control and Computation Engineering
Nowowiejska 15/19, 00-665 Warsaw, Poland
tomasz.winiarski@pw.edu.pl

Abstract. *Social and assistive robotics have vastly increased in popularity in recent years. Due to the wide range of usage, robots executing such tasks must be highly reliable and possess enough functions to satisfy multiple scenarios. This article describes a mobile, artificial intelligence-driven, robotic platform Rico. Its prior usage in similar scenarios, the number of its capabilities, and the experiments it presented should qualify it as a proper arm-less platform for social and assistive circumstances.*
Keywords: *artificial intelligence, mobile robot, social and assistive robotics*

1. Introduction

The ongoing demographic shifts require the development of novel technologies to assist a significant number of elderly or disabled individuals. This demand is met using assistive and social robots [1]. The development of artificial intelligence techniques, particularly efficient audio processing in the computational cloud [2], enables the successful implementation of a voice communication interface as the primary HMI for this class of robots.

When testing the performance of an assistive robot, it is important to consider relevant scenarios. During AAL Incare project[1], Integrated Solution for Innovative

[1] `http://aal-incare.eu/`

DOI:10.17388/WUT.2024.0002.MiNI

Elderly Care, we have tested two potential tasks: assisting with item transportation[2] and reaction to a human fall[3]. For this purpose, we used an early version of the PAL TIAGo-based Rico robot [3]. The robot was also used for experiments with human-aware navigation [4]. These experiments led to the discovery of potential problems arising from the current platform design.

The above experiments resulted in improvements to Rico's robot design and software, briefly described in the following article. We improved the robot's communication capabilities by an additional microphone, implementing active listening behaviour [5] and using artificial intelligence for speech recognition, analysis and synthesis [6]. The addition of a thermal imaging camera [7] made it possible to detect dangerously hot objects. The tactile sensing table [8] allows automatic detection and analysis of objects placed on the robot. Finally, the platform's manual control capabilities have been improved using an additional camera and a smartphone-based GUI [9].

The article starts with a description of the Rico robot in its current state (sec. 2). This robot (or parts thereof) was used in experiments described in sec. 3. The paper is finalised with conclusions (sec. 4).

2. Rico robot

Rico is an extended version of a TIAGo robot [10]. It is a ROS-based mobile robot used in multiple research projects. Its software includes basic ROS packages developed by PAL for the TIAGo robot extended by our platform-specific libraries. Therefore, we decided to employ the MeROS metamodel [11][4] for its specification. Depending on the situation, tasks commissioned for social robots may require appropriate scheduling, including interrupting, postponing and resuming. This led to using the TaskER framework [12] to control the robot's behaviour by switching between tasks dependent on their priority. Due to the possibility of damage to the robot, testing its capabilities is preceded by experiments performed in a simulation environment, making it a simulation-physical system. This means that the same software must be able to run on both the real and simulated versions of the platform. In order to quantitatively evaluate this system, we used the SPSysML [13] philosophy.

[2] `https://vimeo.com/670252925`
[3] `https://vimeo.com/670246589`
[4] `https://github.com/twiniars/MeROS`

The previous version of the robot [3] was extended by additional hardware. We managed to fit two extra sensors inside the robot's head - a directional microphone and a thermal camera (fig. 1). The reasons for such placement were keeping the integrity of the robot's design and the possibility of using them without turning the whole platform around, but only its head.

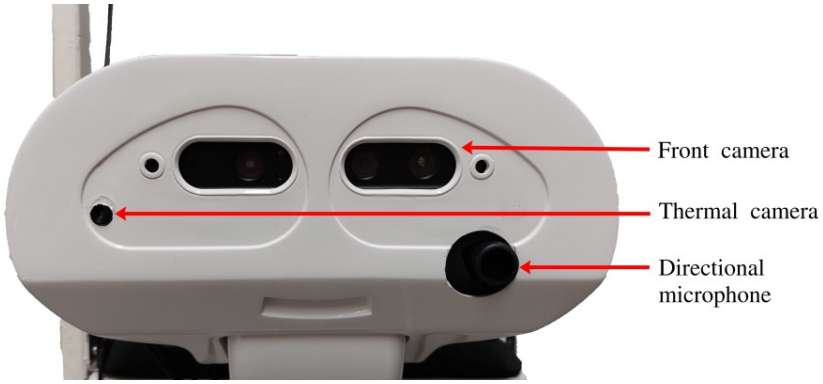
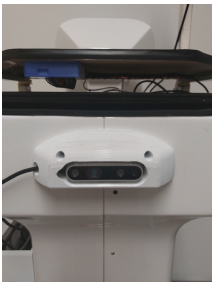


Figure 1: Hardware in Rico's head

In addition to the directional microphone, we kept the omnidirectional one mounted on the pole above the robot's head (fig. 2a). The tactile sensing table was placed on the robot's top, above the original surface (fig. 2b). It includes more than 200 tactile tiles capable of detecting the pressure on their surface. We also left enough space under the table to accommodate the laptop used to process the data from some of the sensors. The final improvement included the addition of an RGBD realsense camera on the back of the robot (fig. 2c). This camera provides high-quality images taken from the opposite side than the camera in the head, which may be helpful, e.g., while manually controlling the robot (for example, through the mobile app). Due to the rising amount of wiring and additional controllers required for all of the new components we attached a special pocket to the side of the robot (fig. 2d). The arrangement of Rico's hardware is presented in fig. 3.

(a) Omnidirectional microphone

(b) Tactile sensing table with laptop shelf below



(c) Rear RGBD camera

(d) Pocket for cable management

Figure 2: New hardware attached to Rico's body

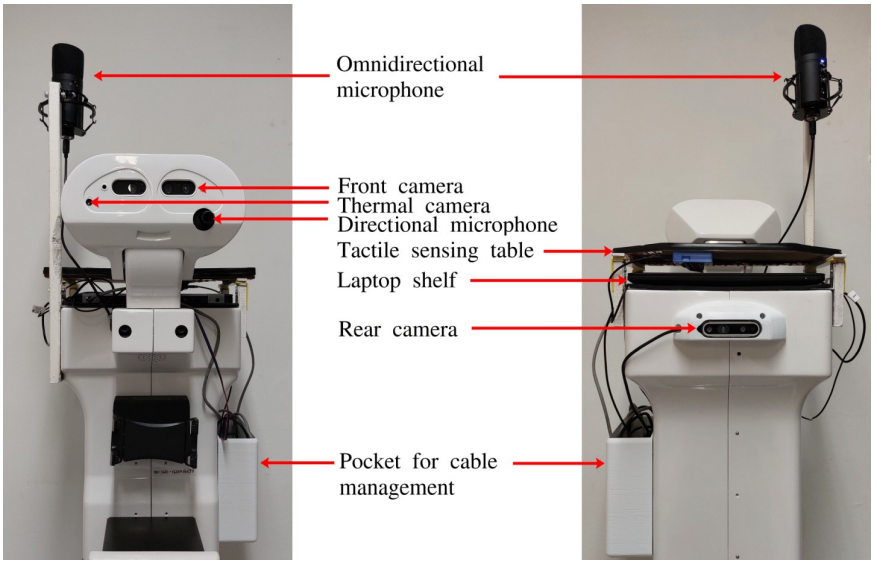


Figure 3: All of Rico's hardware

3. Experiments

The first pool of experiments presented here was designed to address user's talk comprehension issues regardless of distance and orientation relative to the robot [5]. By recording the user with both omnidirectional and directional microphone and processing both inputs simultaneously, we aimed to improve the robot's understanding of voice commands. By communicating with a robot from different places, we were able to examine the quality of the recordings and verify the probability of understanding them. The result of the experiments was to get the robot to understand the command correctly more than 90% of the time. Other behaviour was also developed in which the robot moves to the speaker in case of misunderstanding the command and faces them to better utilize the front microphone [5].

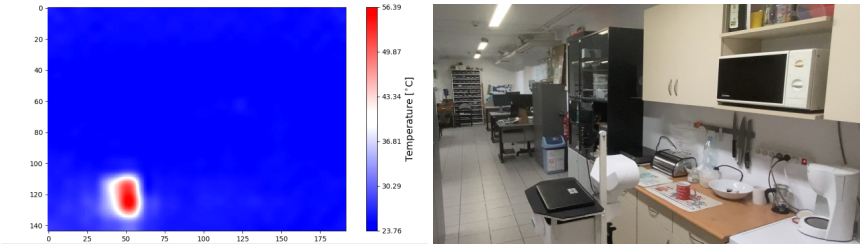
Subsequent experiments were performed to improve the robot's conversational capabilities during task execution [6]. This was done using the OpenAI GPT-3.5 model, which allows for greater flexibility in user commands compared to the previously used Google Dialogflow. In addition, it enabled the object transport scenario to be configured to learn new task parameters and actively use this knowledge in subsequent calls. For example, if a question is asked about the amount of sugar when delivering tea, Rico will start asking for it every time it is asked for tea[6].

Installing a thermal imaging camera made it possible to detect ambient elements that stand out at higher temperatures[7]. During the experiments, detecting a cup filled with hot liquid was undertaken (fig. 4a). The detection of human body temperature from various distances was also tested (fig. 4b). A new scenario was also implemented: the robot patrolled a room by moving between two points. If a hot object was detected, Rico would go to the base to communicate the hazard[7].

To further refine the object transport scenario, experiments using a tactile sensing table were performed [8]. Its readings were used to detect the object's type, position and weight. During the exemplary transport experiment, if Rico detected that the mug was placed too close to the edge or its contents did not match the expected weight, the robot reported the problem to the user present[8].

The last part of the experiments described here involved verifying a smartphone GUI created to manually control the robot [9]. Although not crucial for previously proposed scenarios, manual control is useful, e.g., when autonomous

[5] https://vimeo.com/603908325
[6] https://vimeo.com/863071575
[7] https://vimeo.com/836200374
[8] https://vimeo.com/903840750



(a) Detection result for cup with hot tea (b) Robot position in environment

Figure 4: Heat detection using thermal camera

navigation is unavailable (for example, during the development of the robot's applications). The created interface is presented in fig. 5. During experiments, it was verified that the interface works properly, allowing for movement control of both the base and the head and for visualizing data from all of the robot's cameras and a lidar sensor at satisfactory quality [9].

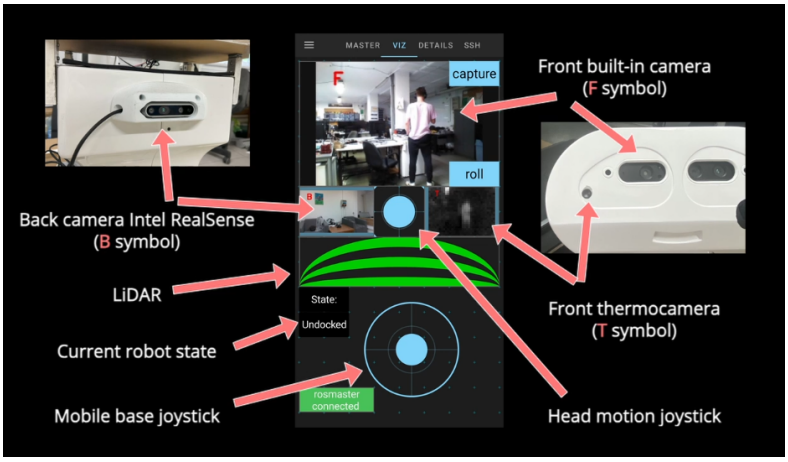


Figure 5: Mobile interface for robot teleoperation

[9] https://vimeo.com/865339765

4. Conclusions

Rico is a mobile arm-less robot designed to carry out usage scenarios in social and assistive applications, thanks to many sensors implemented. It is currently under heavy development to assure reliability and extensive functionality. It uses multiple artificial intelligence algorithms, including a language model, to improve the conversational system and multiple neural networks for detecting and recognising objects placed on the tactile sensing table. We are integrating every developed aspect of the robot's system to create a final version with exhaustive capabilities.

Acknowledgment

The authors would like to thank Jakub Sikora for improving Rico's understanding of speech, Dominika Zając and Stanislau Stankevich for expanding Rico's conversational abilities, Tomasz Indeka and Arpad Micor for installing additional non-standard sensors (tactile sensing table and thermal camera respectively), Kacper Bugała for creating a mobile application for robot's control and installing additional hardware in the form of a rear camera.

The robot was developed as a successor to the INCARE project AAL-2017-059 "Integrated Solution for Innovative Elderly Care". The research was funded by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

References

[1] Leite, I., Martinho, C., and Paiva, A. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.

[2] Recchuto, C., Gava, L., Grassi, L., Grillo, A., Lagomarsino, M., Lanza, D., Liu, Z., Papadopoulos, C., Papadopoulos, I., Scalmato, A., et al. Cloud services for culture aware conversation: Socially assistive robots and virtual assistants. In *2020 17th International Conference on Ubiquitous Robots (UR)*, pages 270–277. IEEE, 2020.

[3] Winiarski, T., Dudek, W., Stefańczyk, M., Łukasz Zieliński, Giełdowski, D., and Seredyński, D. An intent-based approach for creating assistive

robots' control systems. *arXiv preprint arXiv:2005.12106*, 2020. doi: 10.48550/arXiv.2005.12106.

[4] Karwowski, J. and Szynkiewicz, W. Quantitative metrics for benchmarking human-aware robot navigation. *IEEE Access*, 11:79941–79953, 2023. doi: 10.1109/ACCESS.2023.3299178.

[5] Sikora, J. *Voice communication with the social robot Rico*. Master's thesis, WEiTI, 2021.

[6] Stankevich, S. *Conversational system for service robot*. Bachelor's thesis, WEiTI, 2023.

[7] Micor, A. *Detection of temperature hazards using TIAGo social robot*. Bachelor's thesis, WEiTI, 2023.

[8] Indeka, T. *Tactile sensing table in the tasks of an assistive robot*. Master's thesis, WEiTI, 2024.

[9] Bugała, K. *Assistive robot teleoperation*. Bachelor's thesis, WEiTI, 2023.

[10] Pages, J., Marchionni, L., and Ferro, F. Tiago: the modular robot that adapts to different research needs. In *International workshop on robot modularity, IROS*, volume 290. 2016.

[11] Winiarski, T. MeROS: SysML-based Metamodel for ROS-based Systems. *IEEE Access*, 11:82802–82815, 2023. doi:10.1109/ACCESS.2023.3301727.

[12] Dudek, W. and Winiarski, T. Scheduling of a Robot's Tasks With the TaskER Framework. *IEEE Access*, 8:161449–161471, 2020. doi:10.1109/ACCESS.2020.3020265.

[13] Dudek, W., Miguel, N., and Winiarski, T. SPSysML: A meta-model for quantitative evaluation of Simulation-Physical Systems. *arXiv preprint arXiv:2303.09565*, 2023. doi:10.48550/arXiv.2303.09565.

Interpreting and learning voice commands with a Large Language Model for a robot system

Stanislau Stankevich[1][0009−0007−2432−4500],
Wojciech Dudek[1][0000−0001−5326−1034]

[1]*Warsaw University of Technology*
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-662 Warsaw, Poland
wojciech.dudek@pw.edu.pl

Abstract. *Robots are increasingly common in both industry and daily life, such as in nursing homes where they can assist staff. A key challenge is developing intuitive interfaces for easy communication. The use of Large Language Models (LLMs) like GPT-4 has enhanced robot capabilities, allowing for real-time interaction and decision-making. This integration improves robots' adaptability and functionality. This project focuses on merging LLMs with databases to improve decision-making and enable knowledge acquisition for the request interpretation problems.*
Keywords: *Service robots, ROS, LLM, Voice control, intent detection, slot filling*

1. Introduction

We observe the necessity for user-friendly interfaces for seamless human-robot interaction, with voice interfaces being a prime example [1]. The use of Large Language Models (LLMs) like GPT-4 has enhanced robot capabilities, allowing for real-time interaction and decision-making, as demonstrated in [2].

This work seeks to enhance the capabilities of Rico (Fig. 2), a robot developed using PAL Robotics' TIAGo platform [3, 4], designed to assist seniors in care homes. Its example application involves two main human roles: the Senior, an elderly individual who may face mobility or other challenges, and the Keeper, a staff member responsible for assisting the seniors. Rico, the robot, serves as a

DOI:10.17388/WUT.2024.0002.MiNI

third participant, designed to support the Keeper by taking on simpler tasks, e.g. delivering goods to the seniors.

The scenario is an item transportation from the keeper to the senior at the senior's request. However, the item may have various attributes, like tea's temperature, volume, and additional ingredients. The robot must not only fulfil these requests but also learn the available options (like only black coffee being available) by interpreting dialogues.

The process of identifying the user's purpose behind a request is known in NLP as *intent detection*. Additional details included in a request are called *parameters* or *slots* of the intent. For instance, the intent "bring coffee" might include a "size" parameter, while "sing a song" could have parameters like "song name" and "volume". The process of extracting the values of the parameters from user's request is called *slot filling*.

The aim of this work is to enable the system to dynamically expand it's knowledge about given task by identifying and incorporating new intents and slots the during conversations.

The sequence diagram in Fig. 1 presents this use case scenario. The subsequent request for tea should include an immediate tea-type clarification question for the Senior. This learning behaviour should be continued.

2. The conversation system

The system has a ROS-based architecture [5]. It is modelled with MeROS [6] and it is shown in Fig. 3. Each system's component serves a specific function within the system.

User interactions with the robot are handled by the ROS node `Talker`. It converts speech to text and text to speech. This involves turning audio requests into text and the robot's responses into audible speech using OpenAI's models with the web API. The text-converted requests are sent to the LanguageProcessor (`LangProc`) component for further processing.

The ROS node ContextStore (`CS`) records and organises event data during user sessions, like "Rico heard 'bring tea'" or "Rico completed a task." It enables the system to process this information easily, e.g. joining it to prompts for LLM, thus enhancing user interaction.

TaskER is a framework for task management in ROS-based mobile [7] and mobile manipulation [8] robots. It enables task delegation to a robot by sending a

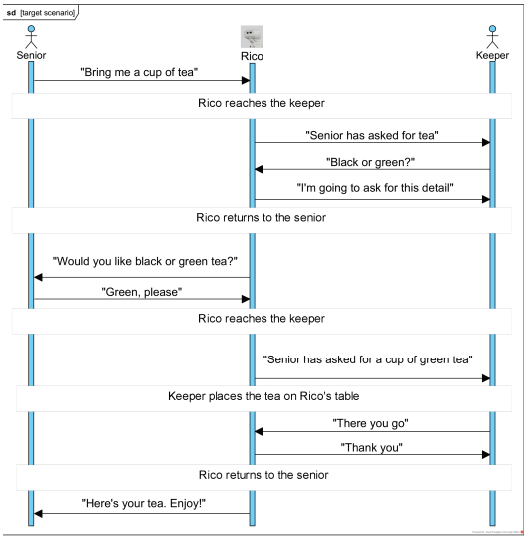


Figure 1: Scenario with bringing an item

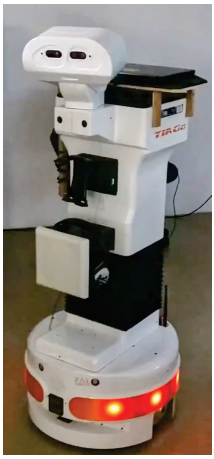


Figure 2: Rico robot

message on a specific ROS topic. the TaskER module encompasses a task repository represented by state machines, defined with SMACH [9].

TaskDatabase (`TD`) is a ROS interface to a PostgreSQL database. It stores known intents along with their parameters and a mapping between intents and tasks that they trigger.

The `LangProc` handles natural language processing in the system using its connection to the LLM (OpenAI GPT-4 API). It crafts prompts for system tasks. For these prompts construction, the `LangProc` queries a database storing known user intents, gets the textual transcription of the user's request, and acquires the current context and scenario (e.g. who is the robot talking with and what was already mentioned in the conversation). Six prompt templates[1] were found to be sufficient for achieving for achieving desired behaviour.

A request for an item (e.g. juice) initiates the application. This request reaches the `LangProc` module, which gathers data about intents known to the system from `TD` and then prompts GPT to guess user intent based on the current state of the

[1] `https://github.com/RCPRG-ros-pkg/rico-language-processor/tree/main/src/stories`

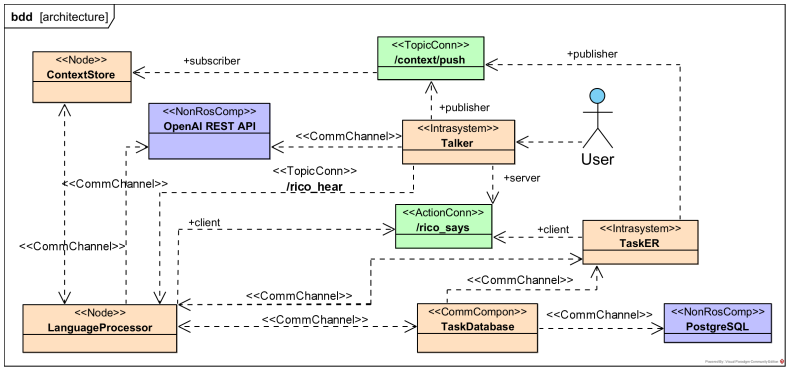


Figure 3: System architecture

conversation. If the system identifies the intent, `LangProc` passes it to TaskER, which then checks `TD` for the related task name and executes the task.

When the robot reaches the kitchen, it asks the keeper for the juice, sending a prompt request to the `LangProc`. `LangProc` fetches the context, queries GPT-4, and the resulting text is vocalized by `Talker`. Throughout, `Talker` updates `CS` with the spoken sentences, extending the system's context.

The question "Which juice?" is set as unexpected by LLM. In consequence, the new intent "Bring juice" with parameter "which" is added to the database, and the robot returns to the senior for the juice type clarifying, as shown in Fig. 4.

Subsequently, `LangProc` again queries GPT-4 with the conversation context, and LLM asks, "What kind of juice would you like?" due to detecting "Bring juice" intent with slot "which" not filled. `Talker` voices this question to the senior, who answers "Apple juice". This expands the context, and LLM extracts all the necessary information, which then reaches the TaskER and the scenario continues as before. The robot reaches the keeper, and GPT-4 generates the robot's initiating conversation with the keeper.

3. Validation and Conclusion

We executed the system in various "bring goods" scenarios, as demonstrated in the video[2]. The system effectively conducted the primary scenario, adeptly

[2] `https://vimeo.com/863071575`

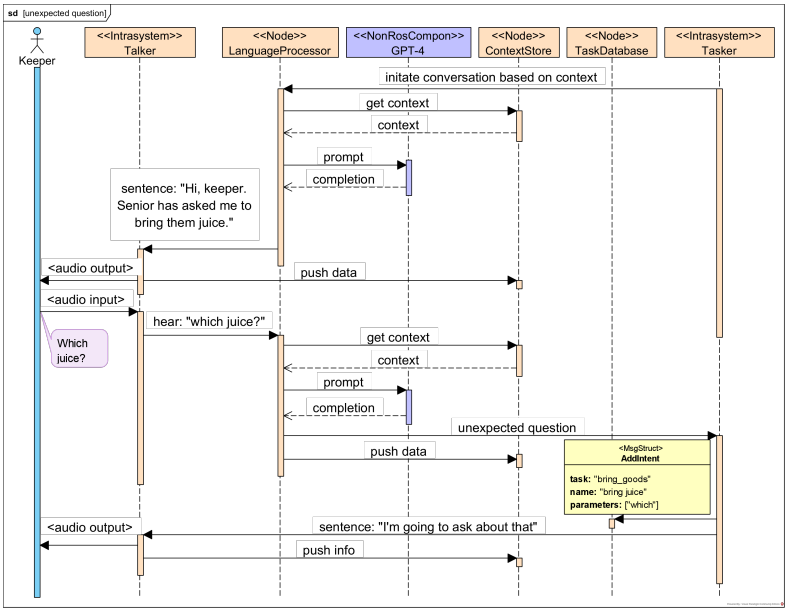


Figure 4: Modules interaction - unexpected question

responding to unexpected inquiries and remembering the "sugar" and "lemon" attributes of the "bring tea" task. However, there is a need to enhance the system's performance. Secondly, the system sometimes struggles with tasks with more than five parameters. We observed that GPT-4 generates inaccurate or "hallucinated" responses, though infrequently. E.g. "Bring me a cup of tea with sugar" may sometimes be misinterpreted, and the "blackOrGreen" slot may be filled as "Black".

In conclusion, we propose an adaptive voice-operated interface for robots. The challenge was integrating the LLM abilities with databases, state machines and the robot controller. Our system uses ROS, speech-to-text and text-to-speech models, and a relational database to deliver vocal conversation functionality with tasks and their parameters' interpretation and exploration. It is usable by a waiter robot, though the system would need customization. Adjusting other Rico's tasks for dynamic intent knowledge expanding requires changing other TaskER state machines by integrating them with `LangProc` and `TD`. The future version could model the robot's environment with LLM prompts and plan tasks by querying LLM.

The most recognized current GPT application in robotics is the Spot robot controlled with voice commands [2]. Our system contrasts with it by extending robot's database using LLMs, instead of providing an API to the robot for LLM. While the idea of LLM-controlled robot gains attention, the robot's ability to explore its environment by using LLM is also desirable.

Acknowledgment

Research was funded by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

References

[1] Seaborn, K., Miyake, N. P., Pennefather, P., and Otake-Matsuura, M. Voice in human–agent interaction: A survey. *ACM Comput. Surv.*, 54(4), 2021. ISSN 0360-0300. doi:10.1145/3386867. URL https://doi.org/10.1145/3386867.

[2] Klingensmith, M., McDonald, M., Agrawal, R., Allum, C., and Shinkle, R. Robots that can chat. *Boston Dynamics Blog*, 2023. URL https://bostondynamics.com/blog/robots-that-can-chat/. Accessed: 2024-02-04.

[3] Pagès, J., Marchionni, L., and Ferro, F. Tiago: the modular robot that adapts to different research needs. 2016. URL https://api.semanticscholar.org/CorpusID:218478582.

[4] Winiarski, T., Dudek, W., Stefanczyk, M., Zielinski, L., Gieldowski, D., and Seredynski, D. An intent-based approach for creating assistive robots' control systems. *CoRR*, abs/2005.12106, 2020. URL https://arxiv.org/abs/2005.12106.

[5] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[6] Winiarski, T. Meros: Sysml-based metamodel for ros-based systems. *IEEE Access*, 11:82802–82815, 2023. doi:10.1109/ACCESS.2023.3301727.

[7] Dudek, W. and Winiarski, T. Scheduling of a robot's tasks with the tasker framework. *IEEE Access*, 8:161449–161471, 2020. doi:10.1109/ACCESS.2020.3020265.

[8] Dudek, W. *Prudent management of interruptible tasks executed by a service robot*. Ph.D. thesis, Warsaw University of Technology, 2021. URL `https://robotyka.ia.pw.edu.pl/papers/phd_thesis_wd.pdf`.

[9] Bohren, J. and Cousins, S. The smach high-level executive [ros news]. *IEEE Robotics & Automation Magazine*, 17(4):18–20, 2010. doi:10.1109/MRA.2010.938836.

Safety of Household Robots - ethical doubts

Artur Gunia[0000−0002−4186−5516]

Jagiellonian University
Faculty of Philosophy
Gołębia 24, 31-007 Kraków, Poland
artur.gunia@uj.edu.pl

Abstract. *The prevalence of household robots is growing steadily year by year. However, in human-robot interaction (HRI), which involves issuing commands to manipulate objects in a given scene, uncertain and dangerous situations may arise. This paper aims to provide a social and ethical viewpoint on the detection of hazardous situation in the scene. I argue that using only machine learning methods may be insufficient to protect ethically questionable situations. To mitigate ethical concerns effectively, a comprehensive approach that integrates multiple modalities and considers contextual nuances is imperative.*
Keywords: *social robots, household robots, anomaly detection, behavior recognition, human-robot interaction (HRI), artificial intelligence*

1. Introduction

The rapid advancement of household robot technology promises widespread affordability and availability soon, facilitating a range of tasks in homes and offices, from cleaning to organizing. As these robots become integral parts of daily life, effective communication between users and machines is crucial, necessitating a foundation of trust in their reliability and safety [1]. For each category of robotic systems, it is essential to establish comprehensive design guidelines that ensure both safety and user-friendliness. For example, the British Standard BS 8611:2023 gives guidance on the identification of potential ethical harm and provides guidelines on the design and operation of robots throughout their entire lifecycle [2]. Furthermore, within forums such as the International Conference Series on Robot Ethics and Standards (ICRES), concerted efforts are made to integrate considerations of robotic safety into the International Organization for Standardization (ISO)

DOI:10.17388/WUT.2024.0002.MiNI

guidelines [3]. Developing ethical considerations in the context of human-robot interactions is crucial, whether in organizational or domestic settings[4].

Currently, the predominant use of household robots is focused on domestic chores, particularly cleaning, with automated vacuum cleaners and mops leading the way. These systems have evolved from hierarchical navigation approaches, first plan, then act [5] - to real-time processing aided by sensor data and machine learning algorithms for object detection [6]. Commands are either preprogrammed, such as pressing a "clean" button, or delivered through basic voice commands like "Start Vacuuming" or "Pause Vacuuming". While integrating vacuum cleaners with intelligent personal assistants like Amazon Alexa and Google Assistant is gaining popularity, the range of available commands remains relatively limited [7].

In the future, household robots are poised to assume more diverse roles beyond cleaning, potentially assisting with shopping tasks, and providing various services [8]. Therefore, they will become more social robots capable of detecting and responding to human emotions [9] underscores the importance of intuitive communication interfaces, emphasizing the need for multimodal systems that integrate verbal and non-verbal (including gestures and gaze) cues [10]. However, this shift towards intuitive communication also presents novel ethical challenges, particularly with respect to inappropriate commands, object recognition, and safe object manipulation by robots.

2. Gap Between Human and Robot Perception

Commanding robots, particularly for physical tasks, poses a significant challenge due to disparities in human and robot perception of the scene. A scene encompasses the environment perceived by agents through receptors or sensors, including visual and auditory inputs. For household robots, the objective often involves directing an agent to move an object from point A to point B within the perceived scene, necessitating a mapping between linguistic cues and visual elements.

While humans find commanding object movements intuitive, programming such actions for robots is complex. Early attempts, such as the SHRDLU system, enabled natural language interactions with computers for manipulating objects, assigning names to groups, and asking about the status of simplified blocks [11]. This model has found applications in computational linguistic research [12] and,

above all, in robotics [13, 14]. However, this system is based on a simplified model of reality, so in a natural conversation over the scene, it may have limitations.

Figure 1. An example of a scene that may pose interpretation challenges.

For effective communication, systems must navigate linguistic and cultural nuances, as ambiguity can hinder task completion and raise ethical concerns. For example (see Fig.1), a command like *"Move the blue mug on the left side of the table next to the yellow mug on the shelf"* may pose interpretation challenges. Thus, bridging the gap between human intent and robot action remains a crucial area of research in robotics.

3. Inappropriate Commands

In an intuitive interaction with a household robot, issuing ambiguous commands can result in the task not being performed correctly. But as a result, unconsciously or even intentionally, the user may issue a command that endangers another person, robot, or property.

Although we, as humans, are eager to use technological conveniences, we often use them inappropriately, dangerously, or deliberately use the systems for illegal activities (an example of which was the interactions of users with the Tay chatbot, which, as a result, began to publish offensive and racist posts on social media). Technology developers must therefore anticipate the possibility of inappropriate behavior on the part of the user, for example, by limiting the functionality of the system in a specific situation [15].

Using current automatic language classification methods, we can detect, for example, hate speech on social networks [16]. Household robots will also need to be equipped with similar modules. Using this solution robot will not execute the following command *"take a knife from the table and stab the person on your left"*, according to the detection of words (such as "knife" and "stab") that can be classified as dangerous terms. But the limitation of this approach is human creativity. The user can formulate commands that, using the above methods, cannot be classified as dangerous, for example the command: *"take an object from the table* ***[knife]*** *and place it facing forward at a height of 1.5 m in the object* ***[in the person at chest height]*** *after your left side."*. This command can be classified as a neutral executable command.

Ensuring safety will not be sufficient using methods based on classification and linguistic analysis (such as keywords or sentiment analysis) regarding issued commands. Therefore, it must include other methods for identifying and interacting with the scene.

4. Dangerous Object Recognition

Ensuring the safety of household robots can be based on computer vision techniques and the classification of objects in the scene. The robot using its RGB camera operates primarily with objects that are in the scene. If certain objects may pose a threat, it is easiest to classify them as those with which the robot will not interact. Hence, it is possible to effectively utilize deep learning models for object detection. Some examples include: *YOLOv7*, *EfficientDet*, *RetinaNet*, *Faster R-CNN*, *Mask R-CNN*. However, the analysis of RGB images will not ensure complete safety in terms of image detection. Apart from the fact of incorrect classification, many other problems are illustrated in Figure 2.

A) The most straightforward approach would involve compiling a list of dangerous objects, such as firearms or bladed weapons, with which household robots should avoid interacting. However, conventional image classification methods may not reliably distinguish between an authentic weapon and a harmless replica. Moreover, an object can be classified as safe by the system, such as an umbrella with a concealed blade, which can potentially serve as a weapon. Rather than relying solely on object classification, an alternative solution could involve assessing the structural characteristics or curvatures of objects to determine their potential danger.

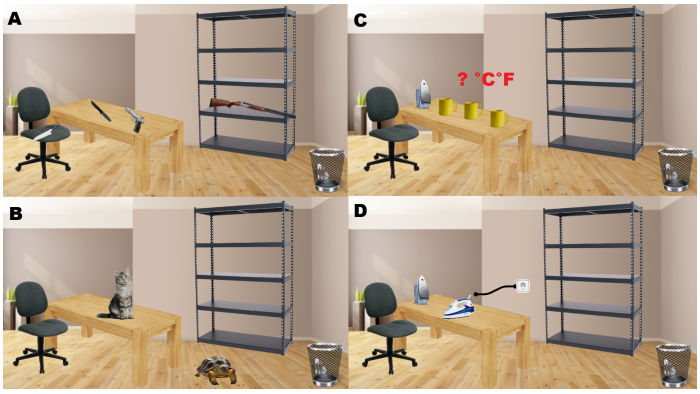


Figure 2. Types of dangers on the scene in visual detection: A) dangerous objects, B) living entities, C) temperature of the objects, and D) connections to other objects.

B) Robot interactions with living entities (excluding humans) should not be allowed. Classification models identify an image so they will not determine whether it is an animal, taxidermy, or animal robot. Detecting whether an object is moving could partially solve this doubt.
C) Sometimes it is not the object itself that is dangerous, but its condition or what is inside it. An example may be a cup that contains hot tea or a dangerous substance such as acid. We cannot determine the condition of a given item based on image classification. Potentially, additional sensors can be used to detect the states of objects, e.g. temperature through infrared cameras, and hazardous substances through the use of spectroscopy techniques.
D) Also, image classification may have a problem in assessing whether objects are connected to each other; a simple example would be an iron that is connected to a socket. The solution will be additional training of models in terms of connections and bonding to other objects.

5. Safe Object Manipulation

The basic structure of a household robot involves three key stages: comprehending the command, recognizing the object, and executing an action on it. Eth-

ical concerns often arise during the final stage, particularly regarding object manipulation and motor functions. Grasping objects poses a significant challenge in robotics, and Convolutional Neural Network (CNN) models such as *GraspNet*, *Con-GraspNet*, and *GGCNN-cl* are commonly used to address this task.

These models focus on how to effectively grab an object, not necessarily on how to move it safely. For example, in human-human interaction, when we want to pass a knife, we grab it by the blade; the robot, in turn, detects such a point in the grip so as not to let go of the object, which does not necessarily have to be safe when moving it. This creates an issue between handling safety and gripping efficiency.

In particular, with regard to object manipulation tasks, it is imperative to consider the trajectory of the robotic entity. Environments where such activities occur are inherently dynamic, predominantly due to human involvement. To mitigate the risk of collisions and ensure safe human-robot interactions, it is crucial to anticipate potential human behaviors. This foresight can be facilitated through the integration of behavior analysis methodologies into robotic systems [17, 18]. Employing RGB and depth cameras, in conjunction with models capable of classifying body (such as *OpenPose*, *PoseNet*) or hand positions (such as *HandPose*, *HandNet*), enables the prediction of human behaviors. Such predictive capabilities are instrumental in identifying potential hazards in the context of human-robot interactions, enhancing safety and efficiency in these environments.

6. Conclusion

Household robots become integral parts of daily life; communication between users and robots must be intuitive and trustworthy. However, challenges arise in ensuring the safe and ethical deployment of these technologies, particularly in scenarios that involve ambiguous commands, identification of potentially dangerous objects, and appropriate handling of objects by robots. While machine learning techniques offer promising avenues for addressing these challenges, a comprehensive approach that integrates multiple modalities and considers contextual nuances is imperative to effectively mitigate ethical concerns.

Acknowledgment

The article is funded by the National Science Center (NCN) in Poland under the project *Intuitive Collaboration with Household Robots in Everyday Settings*. PSP: K/NCN/000142.

References

[1] Gunia, A. The role of trust in human-machine interaction : cognitive science perspective. In M. Sołtysik, M. Gawłowska, B. Śnieżyński, and A. Gunia, editors, *Artificial intelligence, management and trust*, Routledge Studies in Trust Research, pages 85–126. Routledge, New York, 2024.

[2] BS-8611:2023 Robots and robotic devices. Guide to the ethical design and application of robots and robotic systems. British Standards Institution, 2023.

[3] Gwak, K.-W. Service Robot Safety Standardization in ISO. In *Proceedings of the 7th International Conference on Robot Ethics and Standards (ICRES 2022)*, pages 176–178. 2022.

[4] Gunia, A., Sołtysik, M., and Jarosz, S. Robot ethics and artificial morality. In M. Sołtysik, M. Gawłowska, B. Śnieżyński, and A. Gunia, editors, *Artificial intelligence, management and trust*, Routledge Studies in Trust Research, pages 127–143. Routledge, New York, 2024.

[5] Pinheiro, P., Cardozo, E., Wainer, J., and Rohmer, E. Cleaning task planning for an autonomous robot in indoor places with multiples rooms. *International Journal of Machine Learning and Computing*, 5(2):86, 2015.

[6] Ramalingam, B., Lakshmanan, A. K., Ilyas, M., Le, A. V., and Elara, M. R. Cascaded machine-learning technique for debris classification in floor-cleaning robot application. *Applied Sciences*, 8(12):2649, 2018.

[7] Compatible voice commands for google assistant and a wi-fi connected robot, 2023. URL `https://homesupport.irobot.com/s/article/1509`.

[8] Ayub, A., Nehaniv, C. L., and Dautenhahn, K. Don't forget to buy milk: Contextually aware grocery reminder household robot. *2022 IEEE International Conference on Development and Learning (ICDL)*, pages 299–306, 2022.

[9] Błażejowska, G., Gruba, , Indurkhya, B., and Gunia, A. A study on the role of affective feedback in robot-assisted learning. *Sensors*, 23(3), 2023.

[10] Bonarini, A. Communication in human-robot interaction. *Current Robotics Reports*, 1:279–285, 2020.

[11] Winograd, T. Procedures as a representation for data in a computer program for understanding natural language. 1971.

[12] Perera, I., Allen, J. F., Teng, C. M., and Galescu, L. A situated dialogue system for learning structural concepts in blocks world. In *SIGDIAL Conference*. 2018.

[13] She, L., Yang, S., Cheng, Y., Jia, Y., Chai, J., and Xi, N. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 89–97. 2014.

[14] Gao, Y. and Huang, C.-M. Pati: a projection-based augmented table-top interface for robot programming. In *Proceedings of the 24th international conference on intelligent user interfaces*, pages 345–355. 2019.

[15] Gunia, A. and Indurkhya, B. A prototype to study cognitive and aesthetic aspects of mixed reality technologies. In *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, pages 1–6. IEEE, 2017.

[16] Vulnerable community identification using hate speech detection on social media. *Information Processing Management*, 57(3):102087, 2020. ISSN 0306-4573.

[17] Lin, H.-I., Wibowo, F. S., Lathifah, N., and Chen, W.-H. Behavior analysis for increasing the efficiency of human–robot collaboration. *Machines*, 10(11):1045, 2022.

[18] Owaki, K., Techasarntikul, N., and Shimonishi, H. Human behavior analysis in human-robot cooperation with ar glasses. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 20–28. IEEE, 2023.

Impact of planetary rover team composition on mission parameters

Jacek Dobrowolski[1], **Wojciech Brodowski**[1], **Piotr Pałka**[1][0000−0002−0006−363X]

[1] *Warsaw University of Technology*
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
piotr.palka@pw.edu.pl

Abstract. *The use of robotic teams in extraterrestrial planetary missions is a relatively new area of interest for scientists and engineers. Although the idea has not yet been put into practice, research to date indicates that it has great potential. In this paper, we present a description and results of a simulation that determines the speed of mission coverage as a function of time and the number of robots involved. Our multi-agent homogeneous rover system was programmed in a Robot Operating System (ROS) using a modified Jonker-Volgenant algorithm for task assignment.*
Keywords: *multi-agent systems, teams of robots, extraterrestrial planetary missions*

1. Introduction

Space exploration is undoubtedly one of the great technological challenges of modern history. Since the first manned space mission and the moon landing in the 1960s, scientists and engineers have overcome numerous problems to advance man's journey beyond Earth [1], [2], [3]. It was realized fairly quickly that robots are better suited to performing repetitive, laborious tasks, such as collecting soil samples or studying the atmosphere. However, the environment in which they operate is inherently unpredictable. An unexpected disturbance, such as an unknown obstacle or a cosmic ray strike, can cause a serious malfunction.

This is why the concept of using a team of unmanned robots is one of the trending topics in the field [4], [5], [6], [7]. Unlike previous missions, this idea

involves sending a group of robots to specific extraterrestrial destinations - the Moon, Mars or Titan are obviously the most promising. The natural advantage of such a solution is a lower overall failure rate, since the robots are independent of each other from the moment they are deployed on the planet's surface. The robot swarms can be either homo- or heterogeneous, but all of them should follow the common mission objectives predefined by humans. It has been deduced and demonstrated that a team of robots increases important mission parameters such as lifetime and exploration speed.

This paper aims to provide a comparative study of the relationship between mission coverage, number of rovers and total mission time for a team of homogenous rovers. However, this research has the potential to grow and to analyse other, more complex cases, including those consisting of heterogeneous rovers. Our work is simulative, but future plans are based on real experiments.

2. Related work

So far, numerous of research has already been conducted in the field, putting the focus on the following issues. Authors analyse space missions conducted by teams or swarms of rovers [4] or UAVs [5]. Also, teams of homogeneous rovers sharing tasks [4], [8] or heterogeneous specialised robots [6] are considered. The literature proposes various criteria for evaluating missions: speed of terrain exploration, total mission cost, rover lifespan [4], operator's workload [5], or communication capabilities between robots [7]. Also, various algorithms to control and allocate tasks to individual robots are under consideration, such as: a machine learning algorithm for multi-agent systems with high-level human feedback [6]; decentralised, consensus-based task allocation algorithms for multi-agent systems [7]; a traffic planning framework with a dynamic roadmap [8]; a reinforcement learning method for collaborative space exploration [9].

Our approach delves into analyzing planetary mission scenarios in two-dimensional space, following Nayak [8] in terms of methodology and Petrovsky [4] in terms of studied parameters.

3. Research method

Let W be the two-dimensional bounded area. We define P as a known set of points of interest (POI), so that $P \subset W$, and l is the total quantity of POIs. R is a set

of k agents-robots that take part in the mission. Each agent can occupy one of three states: target assigned, target undefined, or mission completed. The objective is to put all the agents in the "mission completed" state, which is only achievable when every point $p \subset P$ has been visited. Each element of P and R sets has its local, two-dimensional coordinates (x, y). In the simulation, we use a modified version of the Jonker-Volgenant algorithm, which itself is a modification of the Hungarian algorithm [10].

Let $C_{k\times l}$ be a cost matrix, which c_{ij} element is a two-dimensional distance of agent i to POI j, defined as $\sqrt{(R_i(x) - P_j(x))^2 - (R_i(y) - P_j(y))^2}$. $X_{k\times l}$ is a resulting assignment matrix, where $x_{ij} = 1$ only if agent i is assigned to POI j. The assumption $\sum_{i=1}^{k} x_{ij} = 1, \forall i \in [1; k]$ states that each agent can only have one POI assigned at any given time. The total cost function results in $min\{\sum_{i=1}^{k} \sum_{j=1}^{l} c_{ij}x_{ij}\}$.

The algorithm starts with calculating the minimum value of each row of the cost matrix. This value is then subtracted from each element of each row. This results in at least one zero element in every row, which simplifies further calculation.

Agents operate on simplified movement and communication principles. They navigate directly to their target POI, which is given to them via a message from the Mission Planner (MP). It has complete information on the location of all rovers and the coordinates of all POIs. To assign POIs to agents to targets mission planner uses a modified Jonker-Volgenant assignment algorithm [11]. Given a set of agents and a set of POIs, it assigns a POI to each agent so that the sum of the distances between them is minimal. Ignoring future target POIs results in POIs being allocated in such a way that a significant distance of useful movement is required to complete the mission. Upon reaching their target POI, agents send a message to the MP, which recalculates and sends new targets to all rovers. We assume that communication in the system is perfect, with no delays, interference or signal attenuation. We also do not consider emergencies such as agent malfunction or unknown obstacles. Agents start on an evenly spaced line in the centre of the test area. They have the same speed of 10 units and the same visibility radius of 50 units. We presume that agents have omnidirectional vision. The total area coverage is measured, defined as $\frac{W_{exp}}{W}$. The agent's vision radius affects W_{exp}, which is the surface of area visited since a start of a mission.

As a backbone of the simulation, we decided to use the Robot Operating System (ROS) – a framework developed to design and manage robot systems. It provides a wide range of possibilities with additional integration with Python and C++ programming environments.

4. Experiments

The goal of the experiments is to test how the number of rovers affects the mission completion time.

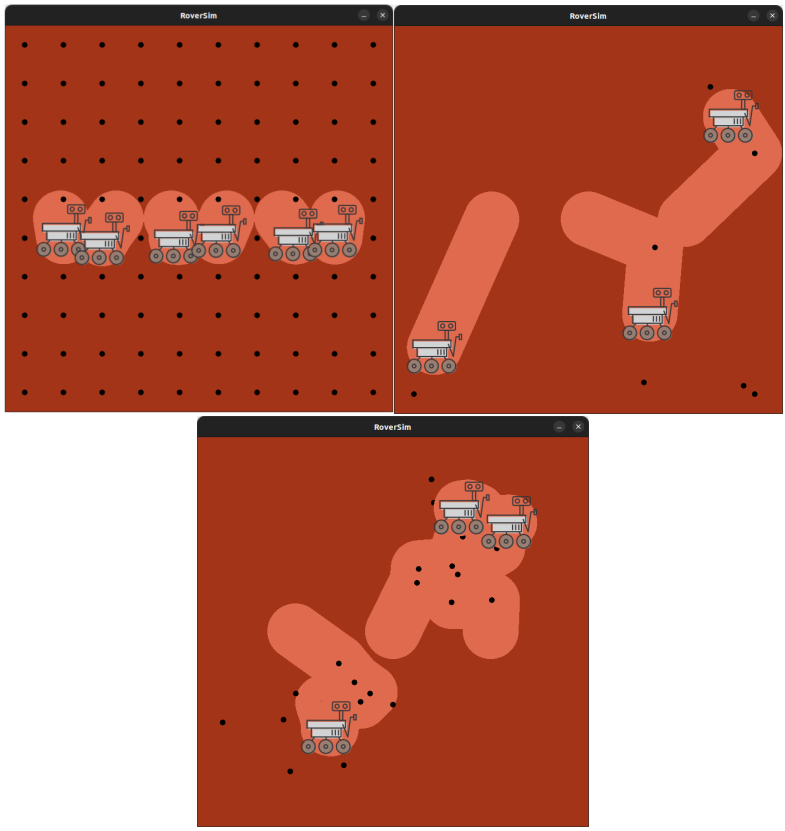


Figure 1. Simulation areas: dense scenario (top-left), sparse scenario (top-right), two clusters scenario (bottom).

Three scenarios have been evaluated: (i) dense POI lattice (see Fig. 1(top-left)) ensures full coverage of the area with rover sensors, 100 POIs; (ii) sparse 7 POI set (see Fig. 1 (top-right)); and (iii) two clusters of POIs located in south-west (10

POIs), and north-east (10 POIs) areas (see Fig. 1(bottom)). In every case, the time to reach all points is measured. The number of rovers ranges from 1 to 6; each simulation configuration was repeated 3 times, and we present the average.

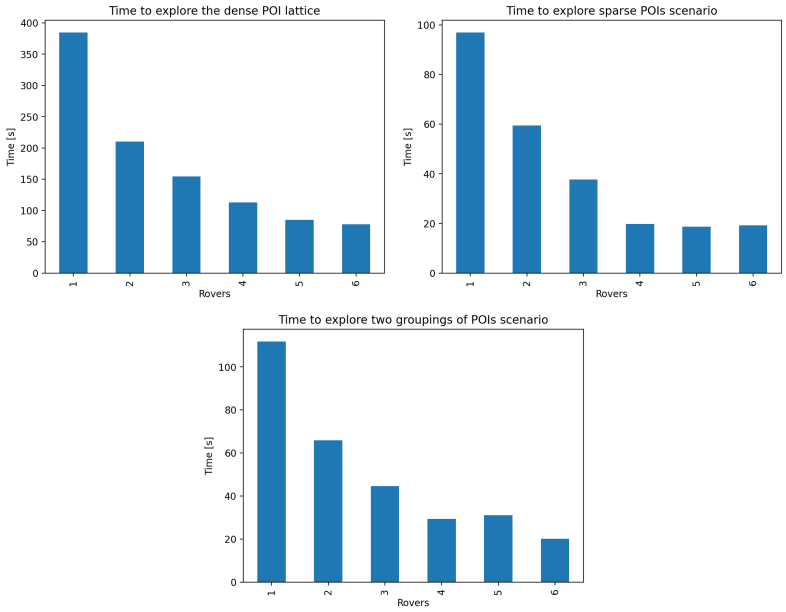


Figure 2. Exploration time in function of number of rovers: dense scenario (top-left); sparse scenario (top-right); two clusters scenario (bottom).

Let us analyse the exploration time against a number of rovers involved in a mission in every scenario. In the dense grid scenario (see Fig. 2(top-left)), time flattens out with five or more rovers. In the sparse grid scenario (see Fig. 2(top-right)), this happens at four, which coincides with the number of POI clusters. This leads to the conclusion that missions with a high number of objectives should consider using a larger number of rovers. For two-clusters case (see Fig. 2(bottom)) time to explore falls up to 4 rovers, but for 5 of them time is greater both in comparison with 4 as well as with 6. This behavior is due to the fact that with an even number of rovers, they distribute equally to each cluster. With an odd number, on the other hand, one of the clusters is reached faster, then at least one of the rovers

has to travel to the other cluster, which increases the time to complete the entire mission.

Missions with a relatively small number of objectives are unlikely to use this large number of rovers. The most significant gain in time is observed with the increase from one to two rovers.

Figure 3 shows how explored area changes in time for various team sizes. In the dense case (see Fig. 3(left)), the larger the team is, the more non-linearities occure in the plot, which shows the redundancy of the rovers in during certain periods of time. All the rovers performed the same tasks (exploration), which proved to be at least partially redundant. A division of labour between the robots would probably result in an optimised use of the available resources. In the two-cluster case (see Fig. 3(right)) for 1 and 2 rovers, the time dependence of the explored area flattens out noticeably, due to the need to travel from one cluster to another. For one rover, this occurs about half the time, for two, toward the end, when one cluster has been explored and the rover goes to help the other complete the mission. For a larger number of rovers, the relationship is close to linear.

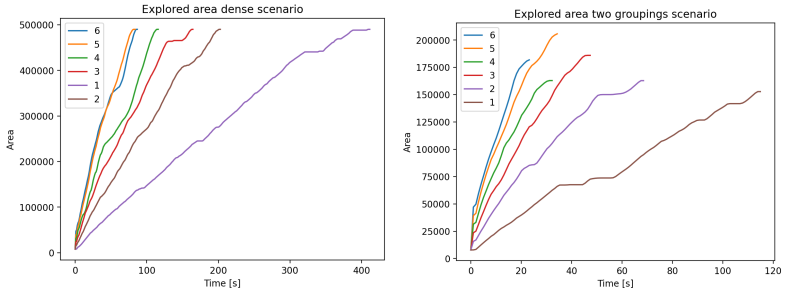


Figure 3. Relation of explored area to exploration time. Dense scenario (left), two clusters scenario (right)

5. Summary

This paper presents a simulation of the exploration of a certain simplified area by a team of homogeneous communicating planetary rovers, with the existence of a Mission Planner that distributes tasks among the individual rovers. Three

scenarios of POI distribution were considered: regular dense, sparse, and two-clusters POIs distribution. We analysed the impact of the number of rovers on the total mission time and the area explored. The results suggest a preferred number of rovers for every scenario.

References

[1] J. P. Grotzinger, J. Crisp, A.R. Vasavada et al. "Mars Science Laboratory Mission and Science Investigation" Space Sci Rev 170, 5–56 (2012).

[2] K. A. Farley, , K. H. Williford, K. M. Stack et al. "Mars 2020 Mission Overview", Space Sci Rev 216, 142 2020.

[3] M. Shirley, E. Balaban "An Overview of Mission Planning for the VIPER Rover.", SpaceOps 2022.

[4] A. Petrovsky, I. Kalinov, P. Karpyshev, D. Tsetserukou, A. Ivanov, "The two-wheeled robotic swarm concept for Mars exploration", 2022

[5] D. St-Onge, M. Kaufmann, J. Panerati, B. Ramtoula, Y. Cao, E. B. J. Coffey, G. Beltrame, "Planetary exploration with robot teams", 2019

[6] M. Colby, L. Yliniemi, K. Tumer, "Autonomous Multiagent Space Exploration with High-Level Human Feedback", 2016

[7] S. Nayak, M. Paton, M. W. Otte, "A Heuristic-Guided Dynamical Multi-Rover Motion Planning Framework for Planetary Surface Missions", 2023

[8] S. Nayak, S. Yeotikar, E. Carillo, E. Rudnick-Cohen, M. K. M. Jafar, R. Patel, S. Azarm, J. W. Herrmann, H. Xu, M. Otte, "Experimental Comparison of Decentralized Task Allocation Algorithms Under Imperfect Communication", 2020

[9] Y. Huang, S. Wu, Z. Mu, X Long, S. Chu, G. Zhao "A multi-agent reinforcement learning method for swarm robots in space collaborative exploration." (2020) 6th international conference on control, automation and robotics (ICCAR) (pp. 139-144). IEEE.

[10] H. W. Kuhn, "The Hungarian method for the assignment problem", 1955

[11] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," in IEEE Transactions on Aerospace and Electronic Systems, vol. 52, no. 4, pp. 1679-1696, August 2016

Slope Estimation for Obstacle Detection in Simultaneous Localization and Mapping

Kürşat Kömürcü[0009−0006−1149−8686], **Linas Petkevičius**[0000−0003−2416−0431]

Vilnius University
Institute of Computer Science, Vilnius University,
Didlaukis str. 47, LT-08303 Vilnius, Lithuania
Vilnius Gediminas Technical University
Antanas Gustaitis Aviation Institute
Linkmenų g. 28-4, 08217 Vilnius
e-mail: kursat.komurcu@mif.stud.vu.lt, linas.petkevicius@mif.vu.lt

Abstract. *This work presents an integrated approach to obstacle detection for Simultaneous Localization and Mapping (SLAM), leveraging LIDAR, stereo cameras, and laser scan sensors, refined by the RANSAC algorithm. Developed using C++, PCL, and ROS Noetic, our system demonstrates enhanced detection capabilities in dynamic environments, validated through our achievement in the Kangal Autonomy Competition. The research is accompanied by the data and reproducible code at Github repository.*
Keywords: *SLAM, LIDAR Mapping, Obstacle Detection, Point Cloud, Slope Estimation*

1. Introduction

The development of Simultaneous Localization and Mapping (SLAM) technologies, notably through the use of LIDAR, stereo cameras, ultrasound, and laser scan sensors, has been instrumental in advancing autonomous navigation by enabling systems to dynamically map and locate themselves within environments [1]. LIDAR sensors, recognized for their accuracy in measuring distances and creating detailed 3D maps, are fundamental for static obstacle detection [2]. Concurrently, stereo cameras contribute to dynamic obstacle detection by estimating slopes and leveraging image disparity for depth and movement analysis [3]. Additionally, the

integration of laser scan sensors enhances obstacle detection capabilities beyond the scope of LIDAR and stereo cameras by analyzing point cloud data derivatives, thus identifying a wider range of obstacles [4].

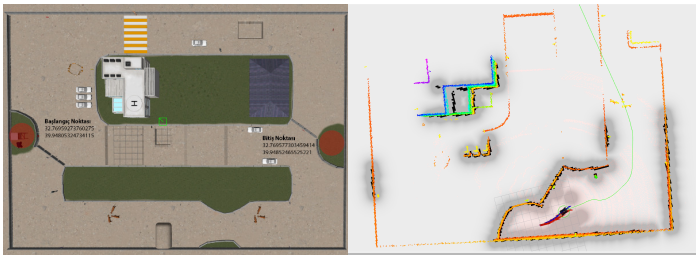


Figure 1. Experiment Map and Simulation of Autonomous Driving

However, while these technologies represent significant advancements, research indicates a need for more integrated and noise-robust solutions for obstacle detection [5, 6]. This study addresses such gaps by proposing a unified sensor fusion approach that combines the strengths of LIDAR, stereo cameras, and laser scan sensors. Through the application of Random Sample Consensus (RANSAC) segmentation on point cloud data, our method not only refines obstacle detection but also improves accuracy in obstacle localization [7], filling a critical void in existing methodologies.

2. Methodology

2.1. Slope Estimation

Integrating both LIDAR and depth camera technologies significantly enhances our autonomous system's obstacle detection and navigation capabilities. This synergistic approach not only capitalizes on the high-resolution obstacle mapping afforded by LIDAR but also harnesses the depth camera's adeptness at identifying dynamic obstacles in close proximity for final navigation setup[1] see Figure 1.

[1]The experimentation setup and algorithm implementation can be access at https://github.com/kursatkomurcu/outdoor_robot Github repository.

2.1.1. LIDAR for Obstacle Mapping

LIDAR sensors are instrumental in developing detailed maps of the environment, accurately capturing static obstacles. The efficacy of LIDAR in mapping and its principles of operation have been well-documented in literature, underscoring its critical role in autonomous navigation systems [8]. The processing of LIDAR data to calculate surface slopes involves analyzing the normal vectors of the points in the generated point cloud. This analysis is pivotal for distinguishing between navigable surfaces and potential obstacles. The slope is calculated as:

$$Slope = \arctan 2(normal_z, \sqrt{normal_x^2 + normal_y^2}) \times \frac{180}{\pi} \quad (1)$$

This methodology for slope calculation based on surface normals is derived from techniques commonly used in digital elevation model (DEM) analysis, highlighting the adaptability of geospatial analysis methods to autonomous navigation [9].

2.1.2. Depth Camera For Dynamic Obstacle Detection

The depth camera is utilized specifically for detecting obstacles that LIDAR may miss, particularly in scenarios involving close-range dynamic objects using obstacle positions and the robot positions. Its real-time responsiveness is crucial for navigating environments with moving obstacles that LIDAR may not effectively detect in immediate proximity [10]. The depth camera's output is processed to calculate the distance of each point in its field of view from the robot's current position:

$$Distance = \sqrt{(x - x_{robot})^2 + (y - y_{robot})^2 + (z - z_{robot})^2} \quad (2)$$

Utilizing this distance measurement, along with the slope information derived from the depth camera's point cloud, we can detect dynamic obstacles. A threshold is set for the slope degree to determine navigability, while also considering the proximity to the robot based on a predefined safety radius [1].

2.1.3. Occupancy Grid Update

Both lidar and camera sensors' data are fused to update an occupancy grid, a discretized representation of the environment where each cell's value indicates the

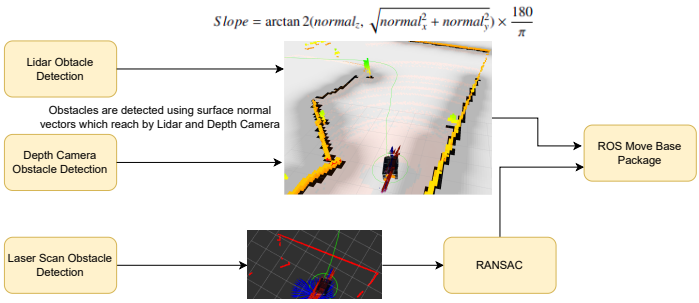


Figure 2. The obstacles assigment to occupancy grid and sensor fusion schema.

presence of an obstacle [11]. The grid indices are calculated as follows:

$$GridX = \lfloor (x - OriginX) \times InvResolution \rfloor$$

$$GridY = \lfloor (y - OriginY) \times InvResolution \rfloor$$

Cells are identified as occupied or free based on a calculation that considers the x and y coordinates of an object, the OriginX and OriginY coordinates of the grid's origin, and the inverse resolution of the occupancy grid in ROS [12]. Occupied cells signal potential hazards outside the robot's safety radius, whereas free cells indicate safe navigation areas. This strategy employs the object's location relative to the grid's origin and the grid's scale to enhance the autonomous system's obstacle detection and navigation [11], ensuring safe maneuverability in varied environments.

2.2. Algorithm and RANSAC for robust estimation

The proposed pipeline first address the problem of address outliers to occupancy grid via slope estimation. The updated grid is combined with other sensors information, the lidar and depth camera. The filtering and combination is finally integrated to the robot's move base package with laser scan data. The full sensor fusion pipeline can be see Figure 2.

Upon gathering and processing the environment data using depth cameras, LIDAR, and laser scan sensors, we implemented the Random Sample Consensus (RANSAC) algorithm to further refine our results. RANSAC is a robust method

for fitting a model to observational data that may contain a significant proportion of outliers [13]. It is particularly effective in environments with a high likelihood of measurement errors or in scenarios where the object's boundaries are not distinctly outlined. In the context of our methodology, RANSAC is employed after the preliminary identification of potential obstacles to distinguish between actual obstacles and spurious data points. The RANSAC algorithm iteratively selects a random subset of the original data points and attempts to fit a model (e.g., a plane or a line) to these points. The fitted model is then used to identify inliers—points that fit the model well within a certain tolerance. The process is repeated a fixed number of times, and the model with the highest number of inliers is selected as the best representation of the data. This experimenting setup is not tested on dynamic obstacles and we will focus on this in future research.

3. Conclusions

Our study has successfully demonstrated the use of a multi-sensor approach, incorporating depth cameras, LIDAR, and laser scanners, to improve obstacle detection in autonomous navigation systems. The integration of slope estimation and the RANSAC algorithm further refined our obstacle detection process, leading to our system's notable performance in the Kangal Autonomy Competition. The research is accompanied by the data and reproducible code at Github repository.

Acknowledgment

We extend our gratitude to the Turkish Defense Industry Presidency and HAVELSAN for supporting and hosting the Kangal Autonomy Competition. Our second-place achievement with Link Robotics was a notable milestone that refined our expertise in autonomous systems.

References

[1] Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics*. MIT Press, 2006.

[2] Goelles, T., Schlager, B., and Muckenhuber, S. Fault detection, isolation, identification and recovery (fdiir) methods for automotive perception sensors

including a detailed literature survey for lidar. *Sensors (Basel, Switzerland)*, 20, 2020. doi:10.3390/s20133662.

[3] Zhang, J. and Lin, X. Advances in fusion of optical imagery and lidar point cloud applied to photogrammetry and remote sensing. *International Journal of Image and Data Fusion*, 8:1–31, 2017. doi:10.1080/19479832.2016.1160960.

[4] Zang, S., Ding, M., Smith, D. B., Tyler, P., Rakotoarivelo, T., and Kâafar, M. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, 14:103–111, 2019. doi:10.1109/MVT.2019.2892497.

[5] Figueroa, F. and Mahajan, A. A robust navigation system for autonomous vehicles using ultrasonics. *Control Engineering Practice*, 2:49–59, 1993.

[6] Pomerleau, D. Neural network based autonomous navigation. pages 83–93, 1990.

[7] Albeaino, G., Gheisari, M., and Franz, B. A systematic review of unmanned aerial vehicle application areas and technologies in the aec domain. *J. Inf. Technol. Constr.*, 24:381–405, 2019.

[8] Li, Y. and Ibanez-Guzman, J. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[9] Smith, M. J., Goodchild, M. F., and Longley, P. A. *Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools*. Matador, 6th edition, 2021.

[10] Rusu, R. B. and Cousins, S. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, 2011.

[11] Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. doi:10.1109/2.30720.

[12] Moravec, H. P. and Elfes, A. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121. 1985.

[13] Fischler, M. A. and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Overlapping Box Suppression Algorithm for Window-Based Object Detection

Aleksandra Kos[0000−0001−9726−4472]

Poznan University of Technology, 60-965 Poznań, Poland
Cufix, 05-825 Grodzisk Mazowiecki, Poland

Abstract. *In this work, we introduce an Overlapping Box Suppression algorithm, which can be used as a building block of any window-based object detection system to inhibit false-positive detections caused by partially visible objects. Window-based approaches are often used for small object detection to make the objects easier to detect by general-purpose detectors. However, this group of methods is typically associated with a reduced detection quality for larger objects due to the intersection of objects with detection windows. The proposed algorithm significantly improves the AP_m by 25% compared to the NMS-based filtering method in our detection system trained on images from the 3D-ZeF20 dataset.*
Keywords: *deep learning, object detection, focus and detect methods*

1. Introduction and Related Work

Since the introduction of deep learning, the quality of object detection has increased rapidly, with modern detectors achieving an Average Precision (AP) of over 50% [1] on the COCO test-dev dataset [2]. Despite the substantial progress made in recent years, the quality of small object detection remains poor, with an AP gap of around 30% between small and large objects [1]. Multiple methods have been developed specifically for detecting small and tiny objects [3], with a significant portion utilizing external guidance to focus the detector on small, relevant regions (windows) cropped from the original, typically high-resolution image. Focus and detect methods [3] increase the relative size of the objects, making their features more visible and easier to detect. However, due to frequently overlapping detection windows, it can also result in a decrease in AP for larger objects, caused by false-positives triggered by partially visible objects. A common

DOI:10.17388/WUT.2024.0002.MiNI

approach in object detection is to use Non-Maximum Suppression (NMS) to filter redundant bounding boxes. However, in window-based methods, partial detections may have a small Intersection over Union (IoU) with full detections, so they are not filtered out by NMS.

In this work, we introduce a simple yet effective Overlapping Box Suppression (OBS) module, which can be integrated into any window-based object detection method. This module is a component of a larger system that we are developing [4] for detecting tiny objects in video data. In [5] prediction scores are used to get average boxes based on detections from multiple models. In our approach, we do not average the returned rectangles; instead, we depend on the assumption that there exists a detection closest to the label (as one of the sub-windows typically shows the entire object). We use both the detection bounding boxes and the coordinates of sub-windows to reduce false-positives resulting from partially visible objects. To simplify visualization and improve result interpretability, we demonstrate our algorithm on the 3D-ZeF dataset [6]. Of all available labels, we only use 2D bounding boxes.

2. Overlapping Box Suppression

We propose the OBS algorithm to mitigate false positives in windows-based object detection systems. These false positives result in low AP, especially for larger objects that are cut by the detection windows. The OBS is employed during the inference stage after the detected objects are aggregated from all sub-windows. To demonstrate the advantages of OBS over NMS in window-based detection, we utilize our object detection system discussed in [4]. This time, however, we integrate a tracking module to obtain RoIs from an additional source. For simplicity, we present the results using the 3D-ZeF20 dataset, which contains few objects in a single image.

2.1. Tiny Object Detection System

Our object detection system incorporates a Region of Interest (RoI) prediction stage into the object detection pipeline. Then, based on the returned RoIs, detection is performed at full resolution within selected fixed-sized image sub-windows. RoIs come from two sources: a binary segmentation network that filters out background regions in the current frame, and a SORT-based [7] object tracker that predicts objects' bounding boxes based on past locations. After performing detections

on independent sub-windows, the detections from these windows are aggregated and transformed into the coordinate system of the original image. Previously [4], we relied solely on the NMS algorithm to filter out redundant detections resulting from overlapping detection sub-windows. In this article, we replace NMS with OBS, which additionally takes into account the coordinates of detection windows and efficiently eliminates false positive detections, a common issue in window-based detection methods.

2.2. Overlapping Box Suppression Algorithm

We identify redundant detections by defining false positives as follows: if a detection $d^i_{w_j}$ detected in a sub-window w_j has a high overlap with the intersection of detection $d^i_{w_k}$ detected in another sub-window w_k with window w_j, then the detection $d^i_{w_j}$ is considered a false-positive. We measure overlap using IoU and

Figure 1. Visualization of how the OBS algorithm works. We show input bounding boxes (D0, D1, D2, D3), IoUs for detections and intersections of detections with detection sub-windows (b-e), and output bounding boxes (D0, D1). W0, W1 - detection sub-windows; I(Wi,Dj) - intersection between sub-window Wi and detected bounding box Dj.

set a threshold for high overlap at 0.7 (this value was selected experimentally). The OBS algorithm takes the detections and the coordinates of the detection sub-windows as inputs. Then, the proposed algorithm computes the intersections of detections and sub-windows and generates an IoU matrix between detections and the previously computed intersections. We identify candidates for filtering using the IoU, we mark for deletion all detections for which the IoU exceeds the threshold for any of the computed intersections. The order of deletion is determined by a cost vector for each candidate. We filter out low score, high IoU, and low area detections first. The operation of the algorithm on sample detections is shown in Fig. 1, where partial detections D2 and D3 are successfully filtered out.

Table 1. Object detection metrics on the sequences from the 3D-ZeF20 dataset. AP - Average Precision, AR - Average Recall; seg - RoIs from the semantic segmentation network, track - RoIs as locations estimated by the tracking algorithm, seg-track - concatenated RoIs from both seg and track. Final image detections were established by concatenating the detections from all windows and filtering using either the NMS or OBS algorithm.

ROI	algoritm	AP	AP_{50}	AP_{75}	AP_m	AP_l	AR_1	AR_{10}	AR_{100}	AR_m	AR_l
seg	NMS	66.7	91.3	72.9	55.4	68.9	**28.7**	**74.3**	**74.4**	**71.9**	**75.4**
	OBS	**67.2**	**92.1**	**73.5**	**58.4**	**69.1**	**28.7**	74.2	74.3	71.8	**75.4**
track	NMS	61.4	84.3	69.1	39.6	65.2	28.0	78.7	80.0	**77.8**	80.9
	OBS	**73.1**	**95.5**	**83.4**	**64.7**	**75.4**	**29.2**	**81.0**	**81.1**	77.0	**82.7**
seg-track	NMS	61.5	84.4	71.0	41.6	65.2	28.1	78.8	80.0	**77.4**	81.1
	OBS	**72.7**	**95.9**	**83.8**	**65.1**	**74.7**	**29.1**	**80.6**	**80.8**	76.6	**82.4**

3. Experiments

To assess the impact of the OBS algorithm on detection quality, we initially established baseline metrics (NMS in Tab. 1) for all three RoI approaches. We then replaced NMS with OBS and recalculated the metrics (OBS in Tab. 1). For the analysis of detection quality, we utilized COCO metrics [2]. However, we excluded both AP_s and AR_s due to the scarcity of small objects in the 3D-ZeF20 dataset. By replacing the NMS with OBS algorithm, AP increased significantly

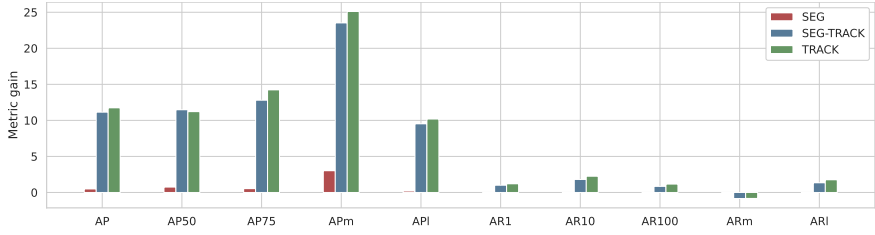


Figure 2. Metric gain caused by replacing NMS with the OBS algorithm.

(Fig. 2) with the most notable increase in AP_m of 25% for the system with RoIs returned by an object tracker. AP increased the most for RoIs given by an object tracker, probably because in this approach there are more detection sub-windows

(as shown in Fig. 3). In a track approach, each false-positive detection leads to the false-positive tracks, and consequently, false-positive sub-windows are propagated to the next frames. OBS does not negatively impact AR, which means that true-

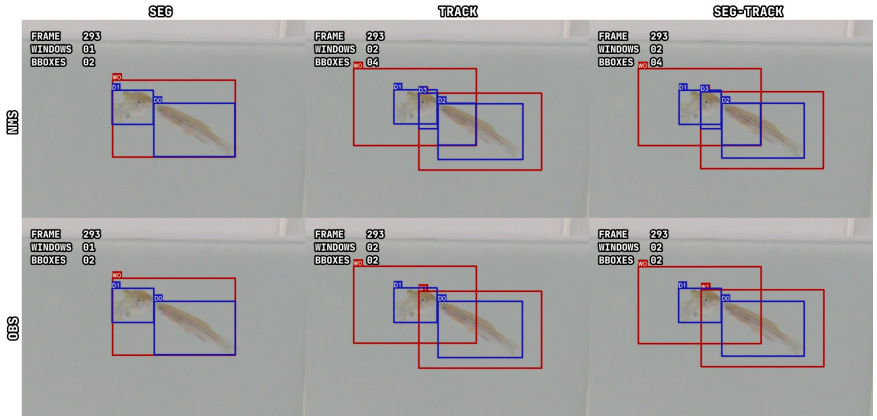


Figure 3. Sample detections for each of the three RoI approaches with NMS (first row) and our OBS algorithm (second row) as the method for aggregating the detections from multiple sub-windows.

positive detections are not filtered out. We only noticed a slight decrease in AR_m, which may be due to inaccurate detections. In Figure 3, both track and seg-track false-positive detections were eliminated thanks to the OBS.

4. Conclusions and Future Work

In this article, we introduced an Overlapping Box Suppression (OBS) algorithm as a post-processing tool for filtering out false-positive detections resulting from partially visible objects. The OBS can be applied to any window-based detection method to address the issue of poor detection quality for larger objects. We demonstrated the effectiveness of the OBS using our tiny object detection system and a simple dataset containing many medium-sized objects. We observed a notable increase in AP_m of 25% for our track-based RoI approach. We plan to test the OBS on other datasets in the future and continue developing additional methods for small and tiny object detection.

Acknowledgment

The research was supported by the Ministry of Education and Science as part of the "Doktorat Wdrożeniowy" program (DWD/5/0203/2021).

References

[1] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475. 2023.

[2] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[3] Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., and Han, J. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[4] Kos, A., Majek, K., and Belter, D. Where to look for tiny objects? ROI prediction for tiny object detection in high resolution images. In *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 721–726. IEEE, 2022.

[5] Solovyev, R., Wang, W., and Gabruseva, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021.

[6] Pedersen, M., Haurum, J. B., Bengtson, S. H., and Moeslund, T. B. 3D-ZeF: A 3d zebrafish tracking benchmark dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2436. 2020.

[7] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

Concurrent real-time optimization in embedded system design process using genetic algorithm

Adam M. Górski[0000−0003−3821−5333],
Maciej Ogorzałek[0000−0003−3314−269X]

Jagiellonian University
Faculty of Physics, Astronomy and Applied Computer Science
prof. Stanisława Łojasiewicza 11, 30-348 Cracow, Poland
{a.gorski, maciej.ogorzalek}@uj.edu.pl

Abstract. *In this paper genetic algorithm for Concurrent real-time optimization in embedded system design process is presented. Generally the optimizing problem is split into two phases. Each one impacts another in real time. The first phase is responsible for a choice of optimizing parameters. The second one optimizes the parameters. The process must include unexpected situations which happen during the work of the system. However many situation can be solved in many ways. Each way could demand execution of different unexpected tasks. Therefore the problem is to establish which way is better before making the choice of the way. Unexpected tasks can change optimizing parameters in real time. Fortunately not every optimizing parameter can be changed. Every ready system is always characterized by some common parameters. The common parameters can help in evaluation of the solutions. Genetic algorithm proposed in this paper will help in finding the optimal combination of unexpected tasks to solve encountered problems by generating some random number of solutions and evolving them using genetic operators: crossover, cloning, mutation and selection.*
Keywords: *genetic algorithm, artificial intelligence, embedded systems, concurrent real-time optimization*

1. Introduction

Embedded system design process can be divided into three parts [1]: modelling, validation and implementation. Górski and Ogorzałek proposed another part of the process – assignment of unexpected tasks [2]. The authors indicated

DOI:10.17388/WUT.2024.0002.MiNI

that during the work of the system some unexpected situations can happen. The system cannot be prepared for every situation. Extending the hardware by new resources can be impossible or too expensive. Therefore the architecture of the system not always can be changed. In [3] authors proposed a genetic programming solution to solve unexpected tasks assign process. The biggest disadvantage of proposed methodologies was that in tasks assignment process every information about unexpected tasks was given. However every unexpected situation can be solved on many ways [4]. Therefore during detecting of unexpected tasks many potential solution can be proposed. Such a process is very dynamical and can be changed in real time. In [5] authors proposed a genetic algorithm for detecting and assignment of unexpected tasks for IoT.

In this paper we present a genetic algorithm methodology for concurrent real-time optimization in unexpected tasks detecting and assignment. Such a problem is a part of embedded system design. Using genetic algorithm [6] it is possible to generate many solutions. The solutions represent many ways of solving the problems. In evolution process all of the individuals are evolved. The paper is organized as follows: the second section includes assumptions, then algorithm is presented. In the next section are results of experiments. In the last section conclusions and directions of future work are presented.

2. Assumptions

2.1. Embedded systems

Embedded system is a computer system microprocessor or microcontroller based optimized to execute some special tasks. Meny of the systems are solved as autonomous ones [7]. Distributed embedded system is consisted of two groups of resources: Processing Elements (PEs) which are responsible for tasks execution and Communication Links (CLs) provide communication and data transfer between tasks. PEs can be divided on: Programmable Processors (PPs) – universal resources able to execute more than one task, and Hardware Cores (HCs) which are specialized resources dedicated to execute only one task. In this paper embedded system is specialized using extended task graph [8] $G = \{T, E\}$. Each node in the graph $T_i \in T$ represents a task. Each edge $e_{ij} \in E$ represents amount of data that needs to be transferred between two connected tasks T_i and T_j. The transmission time is dependent on bandwidth b of the CL and can be calculated

using the following formula:

$$t_{i,j} = \frac{d_{ij}}{b} \tag{1}$$

On a figure 1 below an example of extended task graph is presented. The transmission time is equal to 0 if two connected tasks are executed by the same resource.

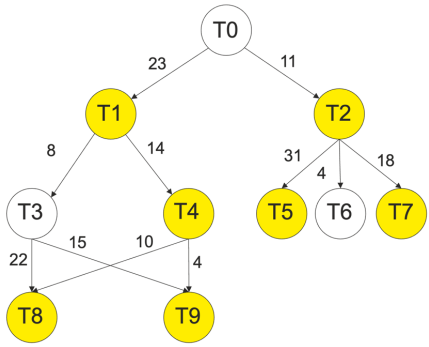


Figure 1. Example of an extended task graph

The graph is consisted of 10 nodes. Tasks T0, T3 and T6 cannot be split into some number of subtasks. Tasks T1, T2, T4, T5, T7, T8 and T9 can be divided into at least two separate subtasks. Each task is characterized by a time (t) and a cost (c) of tasks execution on every PE, The values are given in external database. The goal of a design process is to find the cheapest solution which satisfies time constrains. The overall cost of the system (C_o) is a sum of initial value of a system (C_i) and a cost of unexpected tasks execution:

$$C_o = C_i + \Sigma_{j=0}^{l} C_{T_j} \tag{2}$$

where l is a number of unexpected tasks.

2.2. Picking an apple problem

Picking up an apple can be done in many ways. Every way consists of different tasks. Each way demands different parameters to optimize. It is very difficult to establish which way is better without picking up the apple. The process can be split into two phases. The first one is responsible for choosing parameters to

optimize. The second one makes the optimization of chosen parameters. However depending on the local conditions some of the ways could not lead to success. It is not possible to indicate all of those ways without executing some of the tasks. If during the process it can be found that chosen way of picking up an apple does not lead to the success, or something unexpected happen, the optimizing parameters must be changed. If optimizing parameters are changed the second phase of the process will also be changed. Therefore the phases impact each other in real-time. The solutions are characterized by global optimizing parameters which cannot be changed during the process. The example of such parameters can be: time or cost of the whole process. The final value of the global parameters can be used to compare the results and to choose the best way of picking up the apple.

Such a problem can appear in embedded system design after appearing of unexpected situation. Further information about solving the problem are given in the next section.

3. The algorithm

The unexpected situation can be solved on many ways. Each way is a connection on some number of subtasks. However not every way lead to success. Such an information can be obtained during the unexpected tasks assign process. The process can be consisted of two phases. The first phase is responsible for choosing optimal number of subtasks which can solve unexpected situation. The second phase is responsible for verification of the choice, task assignment and optimization of values of cost and time. Each phase impacts another in real time. The values of cost and time of execution of all the tasks are used to compare the results.

When unexpected situation appears it needed to be inserted into an extended task graph. Presented approach is a genetic algorithm. Therefore at the beginning a randomly generated population needs to be created. Number of individuals created in each population is equal to Π and depends on α parameter given by the designer:

$$\Pi = \alpha * n * p \tag{3}$$

where n is a number of all the tasks in the graph and p is a number of all PEs in the system. The algorithm randomly chooses the subtasks and theirs number. Then chosen subtasks are connected together. If the connection produces appropriate results unexpected tasks are assigned to randomly chosen PEs. The process is

repeated Π times. Next generations are obtained using standard genetic operators: crossover, cloning mutation and rank selection. Number of individuals obtained by the operators are as follows:

$$\Phi = \beta * \pi \tag{4}$$

$$\Psi = \gamma * \pi \tag{5}$$

$$\Omega = \delta * \pi \tag{6}$$

where β, γ and δ control a number of individuals obtained by crossover (β), cloning (γ) and mutation (δ). All of them are given by the designer. The algorithm finishes its work after ϵ generations without better individual.

4. Experimental results

Due to our best knowledge presented methodology is the first which deal with investigated problem in embedded system design. Therefore it is hard to compare obtained results. The algorithm was tested using randomly generated graphs with 10, 20 and 30 nodes. For each set of parameters 30 runs were done. The best and the worst results obtained in the runs are presented in table 1. Control parameters were set as follows: $\alpha = 100, \beta = 0,7, \gamma = 0,1, \delta = 0,2, \epsilon = 5$. It is well known that crossover in genetic algorithms give better results than mutation. Therefore the value of β parameter has the greatest value. Cloning only copies some number of individuals to a new generation therefore value of γ was minimal. The sum of values of β, γ and δ parameters needs to be equal to 1 to have the same number of individuals in generations.

Table 1. Experimental results

Graph	t_{max}	**Best result**			**Worst result**		
		Time	**Cost**	**Generation**	**Time**	**Cost**	**Generation**
10	1600	1537	935	14	1413	1076	11
20	2500	2453	1629	13	2344	2544	15
30	3500	3331	1975	13	2671	2985	18

As it can be observed the algorithm was able to generate valid solution for every tested graph. The difference between the best obtained cost and the worst one for graph with 10 nodes is not too big. It is equal to 141 units. For a bigger graphs

the difference is growing – it is equal to 915 for a graph with 20 nodes and 1010 for a graph with 30 nodes. The difference could be dependent on time constraints, but it is necessary to make more experiments to establish it. The second what can be observed is that with increasing the cost of the system the time of execution of all the tasks is decreasing. It was expected dependency because it is a Pareto optimal problem [9]. When one of the parameters increases its value the second one decreases the value. Such a dependency is not proportional.

5. Conclusions

In this paper a genetic algorithm solution which solve concurrent real-time optimization problem in embedded system design was presented. Proposed algorithm is able to find unexpected tasks and find theirs optimal assignment automatically. Therefore the design process can be more effectively and universal. It is possible to find the solution for more problems. The algorithm also makes the design process cheaper.

In the paper first results were presented. In the future it is planned to test the algorithm on more number of graphs and establish the optimal value of all the parameters. It is also planned to use another evolution algorithm like e.g. genetic programming. Concurrent real-time optimization is quite new problem in operational research. Finding more practical applications and solutions of the problem seems to be a good direction of future work too.

References

[1] De Micheli, G., and Gupta, R., K. Hardware/software co-design. In *proceedings IEEE 95(3)* Mar. IEEE, 1997.

[2] Górski, A., and Ogorzałek, M., J. Assignment of unexpected tasks in embedded system design process. *Microprocessors and Microsystems*, vol. 44, 2016 pages 17– 21. Elsevier.

[3] Górski, A., and Ogorzałek, M., J. Assignment of unexpected tasks in embedded system design process using genetic programming. In *proceedings of the 6th International Conference on the Dynamics of Information Systems (DIS 2023), Lecture Notes in Computer Science*, vol. 14321, 2024, pages 93 – 101. Springer, Cham. https://doi.org/10.1007/978-3-031-50320-7_7.

[4] Górski, A., and Ogorzałek, M., J. Auto-detection and assignment of unexpected tasks in embedded systems design process. In *proceedings of the 23rd International Workshop of the European Group for Intelligent Computing in Engineering*, 2016, pages 179 – 188.

[5] Górski, A., and Ogorzałek, M., J. Concurrent Real-Time optimization of detecting unexpected tasks in IoT design process using GA. In *Late Breaking Papers from the IEEE 2023 Congress on Evolutionary Computation Chicago*, IL, USA, 2023, pages 74 – 77. IEEE.

[6] Mirjalili, S. Genetic Algorithm. *In Evolutionary Algorithms and Neural Networks. Studies in Computational Intelligence*, vol. 780, 2019, pages 43 – 55. Springer, Cham. https://doi.org/10.1007/978-3-319-93025-1_4

[7] Kato, S., Tokunaga, S., Maruyama, Y., Maeda, S., Hirabayashi, M., Y. Kitsukawa, Monrroy, A., Ando, T., Fujii, Y. and Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In *proceedings of 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pages 287–296, IEEE.

[8] Klaus, S. Huss, S. A. and Trautmann T. Automatic generation of scheduled SystemC models of embedded systems from extended task graphs. In *System Specification & Design Languages - Best of FDL '02*, E. Villar and J. P. Mermet, Eds, Kluwer Academic, Norwell, Mass, USA, 2003, pages 207–217.

[9] Mahajan, S., Chauhan, A., and Gupta S., K. On Pareto optimality using novel goal programming approach for fully intuitionistic fuzzy multiobjective quadratic problems. In *Expert Systems with Applications*, vol. 243, 2024. Elsevier. https://doi.org/10.1016/j.eswa.2023.122816

Triplet loss-based metric learning for haptic-only robot localization

Damian Sójka[1,2][0000−0001−9953−1802],
Michał R. Nowicki[1][0000−0002−2299−9025],
Piotr Skrzypczyński[1][0000−0002−9843−2404]

[1]*Poznan University of Technology*
Institute of Robotics and Machine Intelligence
ul. Piotrowo 3A, 60-965 Poznań, Poland
{michal.nowicki,piotr.skrzypczynski}@put.poznan.pl
[2]*IDEAS NCBR*
damian.sojka@doctorate.put.poznan.pl

Abstract. *This study investigates an approach to haptic localization for legged robots, employing triplet loss within a transformer-based neural network. Through experimentation, we evaluate diverse triplet loss variations and their impact on localization accuracy, shedding light on latent space structures. Our findings highlight the superiority of TL-BA triplet loss for haptic-only robot localization, surpassing alternative loss methods. This research not only enhances understanding of machine learning optimization for practical robotics but also identifies effective strategies for haptic localization implementation. Our insights pave the way for more refined methodologies in the development of robotic systems reliant on sparse sensory data.*
Keywords: *localization, haptic sensing, triplet loss, transformers*

1. Introduction

This paper addresses the challenge of efficiently representing haptic signatures in learned prior maps for legged robot localization in extreme environments. While recent research has utilized haptic sensing for this purpose, the problem of efficient representation remains open. We introduce an approach inspired by machine learning trends, combining it with the Monte Carlo algorithm to achieve accurate and computation-efficient localization. Haptic-only localization for legged robots can be of practical value in extreme and dangerous environments, such as those encountered in the mining industry. This localization modality can be used whenever

the external sensors no longer provide useful data, e.g. because of dust or smoke, but a robot equipped with a haptic localization method can return to the starting point in its own footsteps.

Our approach employs the triplet loss concept to learn descriptive embeddings in a transformer-based neural network. We evaluate various triplet loss approaches and elucidate how the efficiency of training with this loss relates to the structure of the latent space in the neural network. Our method removes the need for dense terrain maps, making haptic localization practical for real-world applications. Through experimentation, we illustrate the efficacy of our approach, paving the way for legged robot localization without relying on visual or LiDAR-based SLAM. This research contributes to advancing machine learning for robotic tasks that often use sparse data lacking an intuitive interpretation. Our findings lead to an informed choice of the loss function and provide insights into the dependencies between the latent representation of the sparse data and the learned task performance.

2. Methodology

2.1. Haptic Localization Method

First, we briefly introduce the used localization method depicted in Figure 1. Please see our recent paper [1] for an in-depth description of the localization algorithm and an extended comparison to other haptic localization methods, while this work focuses on the machine learning efficacy side of the problem.

Our method tackles the problem of localizing a legged robot along a known route using only odometry and tactile signals. No prior dense map of the envi-

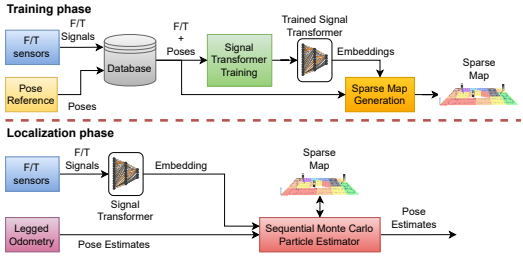


Figure 1: Overview of the training and processing pipelines for our Haptic Localization method [1].

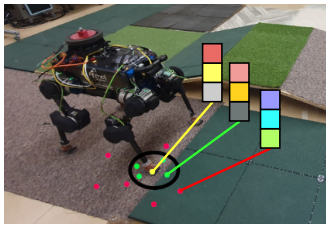


Figure 2: ANYmal B300 robot with a visualization of positive (green) and negative (red) terrain embeddings for the given anchor (yellow).

ronment is required. Instead, only an initial walk along the designated path has to be performed by the robot with a precise source of the robot's pose estimates to gather sparse haptic data. Upon detecting contact between the ground and a foot, the robot records a sequence of readings from force/torque (F/T) sensors embedded in its feet. A single sample from the sequence represents a 6D vector containing 3-axis force and 3-axis torque values. The F/T signals, accompanied by their corresponding probing positions, serve as training data for the neural network. This network is trained to generate compact latent representations of tactile signals, which we term terrain embeddings. Finally, a sparse latent map of the terrain is composed by saving the latent representations of signals on their probing positions.

During the localization phase, we assume a known initial pose. F/T signals are captured in the same manner as before, encoded by the trained network, and compared with embeddings stored in the sparse latent map closest to the currently estimated feet position. The distance between these embeddings in latent space, along with the distance between their estimated probing positions in the real world, is utilized to score sampled positions (particles) within the Sequential Monte Carlo localization algorithm [2]. The scoring process operates under the premise that the greater the similarity between F/T signal embeddings, the higher the likelihood that they originate from the same location. We use Euclidean distance as a distance measure.

It is assumed that the robot's physical properties and gait remain constant from the initial walk to the localization phase, as any alterations could affect tactile readings. The neural network may have difficulty generalizing to substantially shifted F/T signal profiles compared to its training data, potentially leading to reduced localization accuracy. However, this scenario has not been tested yet.

2.2. Learning Terrain Embeddings

The crucial part of our method is a neural network tasked with generating insightful embeddings from haptic signals. These embeddings should encode relevant information required to distinguish between various terrains and locations on the ground. To achieve this, we leveraged a transformer-based encoder network which we called Signal Transformer. Our training strategy draws inspiration from metric learning and place recognition approaches, employing the triplet loss function [3]:

$$\mathcal{L} = \sum_{\substack{a,p,n \\ a \neq p \neq n}} \Big[m + D_{a,p} - D_{a,n} \Big]_+ \tag{1}$$

where $D_{a,p}/D_{a,n}$ is the distance between anchor sample embedding and positive/negative sample embedding, m is a margin hyperparameter, and $[\cdot]_+$ symbol stands for $max(0, \cdot)$ operation.

To determine whether the F/T signal sample x_i belongs to the set of positive samples P_a or negative samples N_a for a given anchor a, we utilize spatial information regarding the locations where the haptic signals were probed. Positive samples encompass signals sampled closer than a defined distance threshold D_{thr} to the anchor. Conversely, negatives are sampled further away than the threshold:

$$\begin{cases} D_{a,i} > D_{\text{thr}} \rightarrow x_i \in N_a \\ D_{a,i} \leq D_{\text{thr}} \rightarrow x_i \in P_a, \end{cases} \tag{2}$$

This concept is depicted in Figure 2. Colored rectangles visualize embedding vectors, where different colors symbolize different vector values.

To avoid the stagnation of training [4], we compose (mine) sets of anchors, positive examples, and negative examples (triplets) online, in each batch separately. Inspired by [4], we use Batch All triplet mining variation in our method by default. Since the F/T signals are not classified in any way, we do not use class-dependent batch sampling and each batch is formed randomly. In Batch All, the loss is averaged over all non-zero loss triplets from the batch.

3. Implementation Details

Experiments are conducted on the PUTany dataset [5]. We evaluate the methods across three distinct sequences (walks) of a quadruped robot ANYmal B300, equipped with F/T sensors in its feet. The sensors capture force and torque values at a frequency of 400 Hz. Consistent with [6], we truncate the signals to include 160 consecutive sensor readings for each step.

As a measure of localization accuracy, we employ the Absolute Pose Error (APE) metric which is calculated as the relative pose between the estimated and ground truth poses averaged over all time steps. Since this study primarily focuses on the latent representation of haptic signals, unlike [1], we exclusively test the variant of our method that employs a sparse latent map with embeddings and 2D positions, without considering the sparse elevation information in the measurement model. By doing so, we aim to minimize the impact of factors unrelated to the embeddings on localization accuracy. However, excluding the elevation data deprives the method of any information concerning the position along the axis perpendicular to the ground. Consequently, pose estimates exhibit drift alongside odometry

in the Z-axis. Therefore, when estimating localization accuracy, we only consider 2D pose within a plane parallel to the ground.

Apart from our default approach of training the network using triplet loss with Batch All triplet mining (TL-BA), we also test Batch Hard triplet mining [4] (TL-BH). Batch Hard involves averaging the loss only over the hardest triplets - selecting only the hardest positives and negatives for each anchor.

Moreover, we conduct experiments with both of the above-mentioned triplet loss variations using a modification in the form of soft-margin (TLS-BA, TLS-BH). The hinge function in triplet loss $[m + \cdot]_+$ is replaced by a smooth approximation $\ln(1 + \exp(\cdot))$. This allows us to pull together the positive pairs as much as possible, without the borderline margin value.

Additionally, for a wider comparison, we consider results obtained with the transformer network trained by a classic contrastive loss [7] (CL), and the reconstruction loss paired with autoencoder architecture from [6] (RL). To verify the neural network architecture influence, TL-BA and TL-BH variations of triplet loss are also tested with the encoder part of the autoencoder architecture (AE-TL-BA, AE-TL-BH).

To ensure a fair comparison among all trained models, we normalized the distances between embeddings in each experiment. It involved dividing them by the maximum possible value before computing the measurement probability for the Monte Carlo Localization algorithm. This procedure guarantees a consistent ratio of influence on particle scores between embedding distances and step location distances across all experiments, thereby neglecting the ratio impact on the experiment outcome.

The results were derived from experiments conducted with varying loss parameters, where applicable: the margin m from equation (1) and the distance threshold D_{thr} from equation (2). These parameters were explored across a grid where m ranges $\{0.01, 0.1, 0.5\}$ and D_{thr} spans $\{0.1, 0.25, 0.5\}$. The most favorable outcomes obtained are presented.

4. Loss Function Impact on Haptic Localization

Table 1 presents the experimental results, including localization accuracy achieved using only the legged odometry estimator Two State Implicit Filter (TSIF) [8] for reference. The TL-BA approach demonstrated the best localization accuracy, affirming the appropriateness of our initial method selection. Embeddings learned through reconstruction loss (RL) exhibit inferior performance in this

Seq.	TSIF	CL	TL-BA	TL-BH	TLS-BA	TLS-BH	AE-RL	AE-TL-BA	AE-TL-BH
1	33.8	7.1	**5.9**	8.3	11.1	7.8	12.3	8.8	8.2
2	92.0	10.8	**7.3**	9.8	15.8	9.3	22.0	9.0	8.6
3	50.7	9.0	**8.7**	11.5	15.0	9.5	13.2	10.6	8.8
Avg.	58.8	9.0	**7.3**	9.9	14.0	8.9	15.8	9.5	8.5

Table 1: 2D Absolute Pose Error (APE) ↓ [cm] results of haptic localization method [1] using different neural network models trained with different losses for terrain embeddings generation.

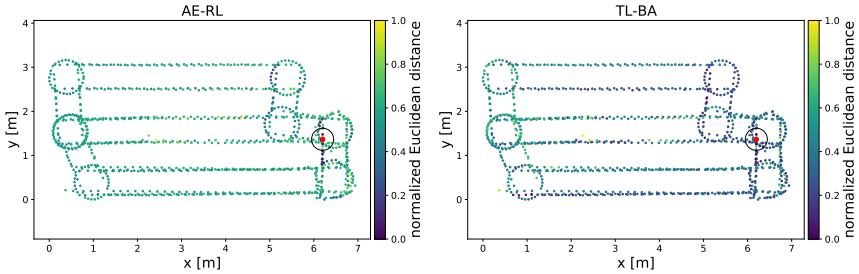


Figure 3: Steps of test sequence 1, where step colors depend on the distance of their signal embedding to the embedding of the red circled step. The left plot depicts embeddings generated with AE-RL model, the right one presents TL-BA model embeddings.

use case, highlighting the advantage of incorporating information about probing locations during training. However, RL can sometimes achieve comparable results to those achieved by using a model trained with the worse-performing contrastive-type loss function. The classic contrastive loss (CL) performs well and may outperform some triplet loss variations. The results with the autoencoder suggest that different triplet loss variations might be more effective depending on the neural network architecture. Pulling the positive samples as much as possible (TLS-BA, TLS-BH) does not result in a performance improvement.

5. Localization Efficacy and Latent Space Structure

Figure 3 displays the step locations of test sequence no. 1 represented as points. The color of each point corresponds to the Euclidean distance between the terrain embedding of the F/T signal probed at each step and the terrain embedding of a signal probed at the location marked by the red circle. On the left plot, the embeddings are based on those generated by the autoencoder trained with

reconstruction loss, while the right one visualizes embeddings generated by the TL-BA configuration. Triplet loss demonstrates the ability to bring embeddings of signals probed closer together in latent space, a feature less evident in the case of AE-RL. This observation suggests that the similarity of the latent space structure and the spatial positioning of steps positively influences the localization results.

6. Conclusion

This study delved into the impact of various triplet loss variations on the localization accuracy of haptic-only robot localization. It compared these results with those achieved using alternative losses and provided insights into the latent space structure obtained by different training methodologies. The experiments indicate that among all tested losses, the TL-BA triplet loss variation yields the best outcomes for this specific task. Moreover, the results offer insights into what might work and what is not likely to be a beneficial strategy for haptic localization implementation.

Acknowledgment

This work was funded by the PUT 0214/SBAD/0242 internal grant.

References

[1] Sójka, D., Nowicki, M. R., and Skrzypczyński, P. Learning an efficient terrain representation for haptic localization of a legged robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12170–12176. 2023.

[2] Buchanan, R., Camurri, M., and Fallon, M. Haptic sequential Monte Carlo localization for quadrupedal locomotion in vision-denied scenarios. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.

[3] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[4] Hermans, A., Beyer, L., and Leibe, B. In defense of the triplet loss for person re-identification, 2017. URL `https://arxiv.org/abs/1703.07737`.

[5] Buchanan, R., Bednarek, J., Camurri, M., Nowicki, M., Walas, K., and Fallon, M. Navigating by touch: haptic Monte Carlo localization via geometric sensing and terrain classification. *Autonomous Robots*, 45:1–15, 2021.

[6] Łysakowski, M., Nowicki, M. R., Buchanan, R., Camurri, M., Fallon, M., and Walas, K. Unsupervised learning of terrain representations for haptic Monte Carlo localization. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4642–4648. 2022.

[7] Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. 2006.

[8] Bloesch, M., Burri, M., Sommer, H., Siegwart, R., and Hutter, M. The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters*, 3(1):573–580, 2018.

Reinforced Learning for Ground Movement of the Hyperreduntant Modular Robot

Jacek Szklarski[0000−0003−1541−9878], **Ela Zawidzka**[0000−0003−1243−9355], **Machi Zawidzki**[0000−0001−8695−4400]

Institute of Fundamental Technological Research
Polish Academy of Sciences
Pawińskiego 5b, 02-106 Warsaw, Poland
jszklar@ippt.pan.pl

Abstract. *We show that the Soft Actor-Critic Reinforced Learning algorithm is able to find efficient motion patterns for a hyper-redundant robot consisting of 6 identical modules connected in a chain like fashion. The control is done by applying relative angular velocities between the modules. Analogous system has been studied before in the context of a robotic trunk-like manipulator.*
Keywords: *Reinforced Learning, Robotic Locomotion, Hyperreduntant Modular Manipulator, Arm-Z, Extremely Modular System*

1. Introduction

Biological snakes are extremely well adapted for different environments. This is mostly the result of the high redundancy of the snake mechanisms. In many instances of irregular environments the bio-inspired robots outperform conventional wheeled, legged or tracked robots. The snake-resembling robots are researched already for a few decades. This type of locomotion has been studied already in the 1940s [1], and a half century later, its rigorous mathematical model has been developed. In the late 90's, a trunk-like locomotors and manipulators have been introduced in [2].

A number of various snake-like robots have been built [3]; most of the designs were intended for crawling on ground [4, 5, 6, 7, 8], some of them for swimming [9, 10], and even fewer for both swimming and crawling on the ground [11, 12].

DOI:10.17388/WUT.2024.0002.MiNI

Snake-like robots are also developed in the context of space applications. In particular, NASAs Exobiology Extant Life Surveyor (EELS) [13] – somehow mechanically similar to the robot discussed here – is being designed for operation in icy crust of Saturn's moon Enceladus (however, EELS, has many potential down-to-Earth application as well).

Fig. 1 shows one of modular snake robots developed in Biorobotics Lab at the Carnegie Mellon University, which is not only capable of crawling on the ground but can also climb vertical posts (for more information see [14]).

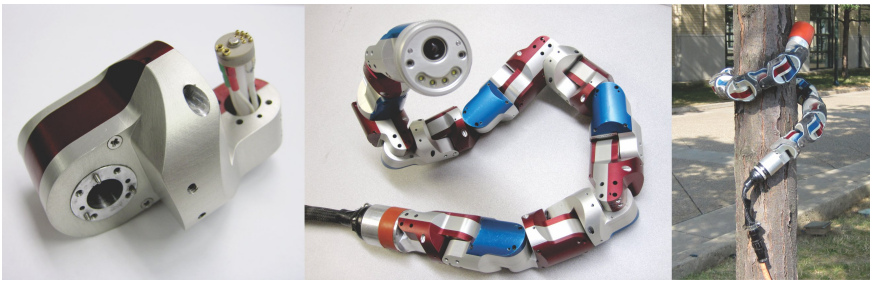


Figure 1. Modular Snake Robot "Uncle Sam". From the left: the module, the robot and the tree climbing action. Photographs ©2010 Biorobotics Lab at the Carnegie Mellon University.

For a concise overview of other biological inspirations for robot concepts see [15].

2. Extremely Modular Hyperredundant Robot

The snake robot presented here is based on *Arm-Z*, which is a concept of robotic manipulator introduced in [16] composed of congruent modules each having one degree of freedom (1-DOF) - a relative twist. In principle, each module is defined by the following parameters: size r, offset d, and ζ, that is the angle between upper (**T**) and lower (**B**) faces of the module. s (slenderness), is an additional parameter, i.e. a d to r ratio. Fig. 2 shows the geometrical interpretation of these parameters and the early, functional prototype.

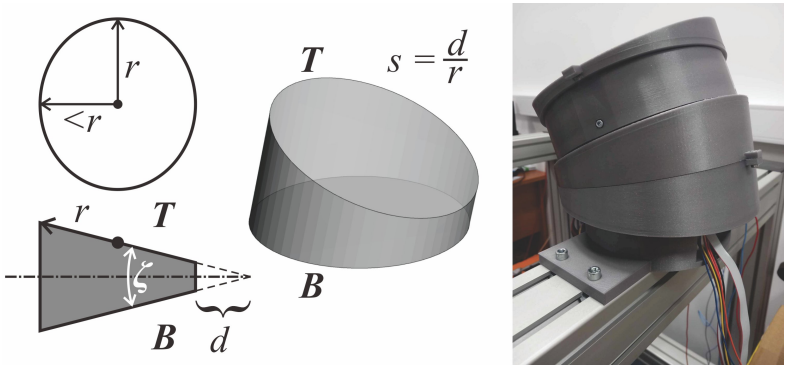


Figure 2. On the left: visualization of the *Arm-Z* unit defined by three parameters: r, d and ζ. On the right: an early functional prototype of the *Arm-Z* manipulator with partial chain of congruent units.

For more information on the concept of *Arm-Z* including its folding study see [17], for early successful implementations of meta-heuristics to control of *Arm-Z* see [18], for documentation of several preliminary prototypes see [19, 20]. The domain of robotic locomotion traditionally uses classical methods designed to address the intricate challenges of enabling robots to navigate their surroundings. Conventional approaches involve specifically engineered control systems and predefined algorithms. However, despite these classical methodologies, locomotion in complex and dynamic environments remains difficult due to the inherent uncertainties and mathematical problems related to kinematic and inverse kinematics. The difficulties in modeling and adapting to diverse terrains, unforeseen obstacles, and dynamic scenarios underscore the need for innovative solutions. Reinforcement learning (RL) emerges as a compelling alternative, offering the capability to learn adaptive locomotion strategies through interaction with the environment and it has been successfully applied in many cases [21]. Due to the non-trival kinematics and possible ways of interaction with the environment, it is tempting to verify if RL can be used to control the hyper-redundant manipulator so as to enable its horizontal movement on the ground.

3. The model and results

Choosing the right physics engine is of fundamental importance for effective RL in robotics. The physics engine acts as the virtual playground for agents to learn and interact, impacting training accuracy and efficiency. For the results presented

here, PyBullet [22] was use due to its accuracy in modeling physical interactions, computational efficiency, and open source nature. The mass of each module is assumed to be 1kg and the radius $R = 1m$. The friction parameters for the flat ground are lateral=1.0, rolling=0.5, spinning=0.9. Using these values PyBullet calculates "real physical" coefficients of friction during contact.

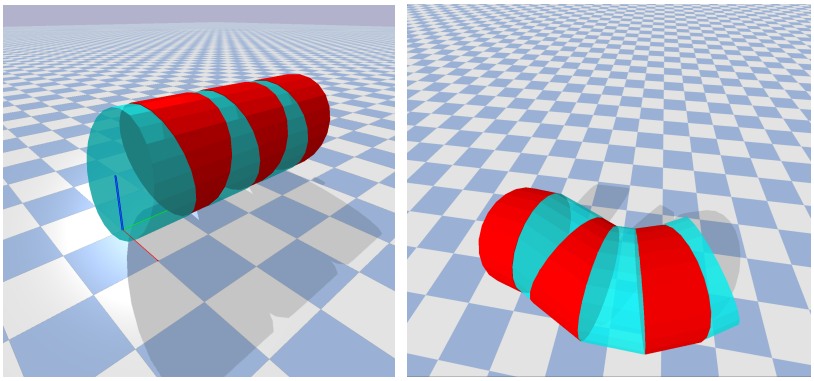


Figure 3. Images rendered by the physics engine representing the robot (alternating colors were chosen to distinguish the modules). Left: the initial state, right the state at timestep=70.

States of the system is described by a vector $S = [s_1, \ldots, s_5]$ in this case, since there are 6 modules and therefore 5 relative twists between them, described as s_i [rad]. The observation of the model is assumed to be simply S. Actions consist of expected angular velocities $A = [\omega_1, \ldots \omega_5]$, the maximum allowed velocity is ± 1rad s^{-1}. The physics engine applies torque in order to achieve the desired angular velocity, the maximum allowed value of torque is 100 [N m].

As the RL algorithm, the Soft Actor-Critic (SAC) off-policy [23] was chosen in the discussed case and its implementation in Stable-Baselines3 [24]. SAC has been successfully applied to a number of interesting problems, including navigation of mobile robots [25]. After basic hyperparameter tuning, the following parameters were applied for learning: learning rate=7.3e-4, batch size=256, γ=0.98, τ=0.02, train freq=8, and SDE were used. The same network architecture (256,256) was used for both the actor and the critic. At each step the physics engine simulates 0.5s during which the torques are applied and contact points and frictions are handled. The reward is simply the sum of the distanced of module centers from the center

of the coordinate system. It rewards strategies generating fast motion in a straight line. Initially the robot lies flat $s_i = \pi$ along the y axis (Fig. 3 left). During the learning process the assumed maximal time horizon is 200 time steps. After about 100k steps of training the model is trained in the sense that the average return does not improve. The model described below is the best in 1M steps (5k episodes).

Figure 4 depicts actions and states in the first 100 steps generated by the model from the initial state. It is evident that - after the initial transition - a rather simple velocity pattern emerges which basically constitutes of two configurations. Switching between these two states leads to motion along almost straight line.

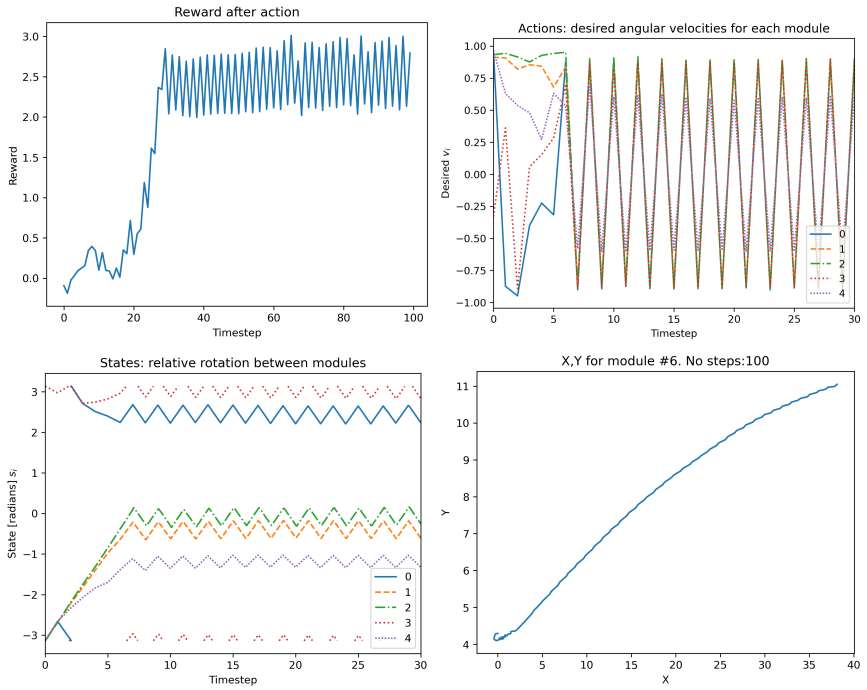


Figure 4. Reward, corresponding actions ω_i, states s_i and a the projection of the module 6 center onto the (x, y) plane during first 30 or 100 timesteps (corresponding to 15 or 50 s).

4. Conclusions

It has been shown that – by means of RL methods – it is possible to obtain motion patterns generating horizontal movement on the ground of the discussed robot. Further study will focus on models with larger number of modules as well as incorporating more control regarding movement direction. Study against module failure will be performed. Current simulations lack any random fluctuations of the environment (except the random nature of the RL control algorithms) which might have significant impact on the effectiveness of the RL algorithms. Our ongoing research is focused on learning with simulations in more complex, random, hilly terrain and additional noisy perturbations in the control process. Additionally, work on physical prototype of the robot will be continued which will also make possible to perform simulations for RL with more realistic physical properties.

Acknowledgments

This research is a part of the project titled *Arm-Z: an extremely modular hyper-redundant low-cost manipulator – development of control methods and efficiency analysis* and funded by *OPUS 17* research grant No. 2019/33/B/ST8/02791 supported by the National Science Centre, Poland.

References

[1] Gray, J. The mechanism of locomotion in snakes. *J Exp Biol.*, 23(2):101–120, 1946.

[2] Hirose, S. *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press, 1993.

[3] Ijspeert, A. J. and Crespi, A. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In *Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA 2007)*, pages 262–268. IEEE, 2007.

[4] Chirikjian, G. and Burdick, J. Design, implementation, and experiments with a thirty-degree-of-freedom "hyper-redundant" robot. *ISRAM 1992*, 1992.

[5] Klaassen, B. and Paap, K. L. GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 4, pages 3014–3019. IEEE, 1999.

[6] Miller, G. Snake robots for search and rescue, neurotechnology for biomimetic robots. In *Proc. Int. Conf. Robotics and Automation (ICRA)*, pages 16–21. 2002.

[7] Choi, H. and Ryew, S. Robotic system with active steering capability for internal inspection of urban gas pipelines. *Mechatronics*, 12(5):713–736, 2002.

[8] Tsakiris, D. P., Sfakiotakis, M., Menciassi, A., La Spina, G., and Dario, P. Polychaete-like undulatory robotic locomotion. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3018–3023. IEEE, 2005.

[9] Melsaac, K. and Ostrowski, J. P. A geometric approach to anguilliform locomotion: modelling of an underwater eel robot. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 4, pages 2843–2848. IEEE, 1999.

[10] Wilbur, C., Vorus, W., and Cao, Y. 14 a lamprey-based undulatory vehicle. *Neurotechnology for biomimetic robots*, 1(9):285, 2002.

[11] Yamada, H. Development of amphibious snake-like robot ACM-R5. In *the 36th International Symposium on Robotics (ISR 2005), Tokyo*. 2005.

[12] Crespi, A., Badertscher, A., Guignard, A., and Ijspeert, A. J. AmphiBot I: an amphibious snake-like robot. *Robotics and Autonomous Systems*, 50(4):163–175, 2005.

[13] Ono, M., Carpenter, K., Cable, M. L., Wilcox, B. H., and Tosi, L. P. Exobiology extant life surveyor (eels). In *AGU Fall Meeting Abstracts*, volume 2019, pages P21D–3410. 2019.

[14] Wright, C., Johnson, A., Peck, A., McCord, Z., Naaktgeboren, A., Gianfortoni, P., Gonzalez-Rivero, M., Hatton, R., and Choset, H. Design of a modular snake robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2609–2614. IEEE, 2007.

[15] Zielińska, T. Transforming biological patterns into robot concepts. *Pomiary Automatyka Robotyka*, 27, 2023.

[16] Zawidzki, M. and Nagakura, T. Arm-z: a modular virtual manipulative. In *Proceedings of the 16th International Conference on Geometry and Graphics*, pages 75–80. 2014.

[17] Zawidzki, M. Deployable pipe-z. *Acta Astronautica*, 127:20–30, 2016.

[18] Zawidzki, M. and Szklarski, J. Transformations of Arm-Z modular manipulator with Particle Swarm Optimization. *ADV ENG SOFTW.*, 126:147–160, 2018.

[19] Zawidzka, E., Kiński, W., and Zawidzki, M. Preliminary prototype of a 4-unit arm-z hyper-redundant modular manipulator. In *Conference on Automation*, pages 285–294. Springer, 2021.

[20] Zawidzka, E., Szklarski, J., Kiński, W., and Zawidzki, M. Prototype of the arm-z modular solar tracker. In *Conference on Automation*, pages 273–282. Springer, 2022.

[21] Zhang, T. and Mo, H. Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems*, 18(3):17298814211007305, 2021.

[22] Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2021.

[23] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[24] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

[25] de Jesus, J. C., Kich, V. A., Kolling, A. H., Grando, R. B., Cuadros, M. A. d. S. L., and Gamarra, D. F. T. Soft actor-critic for navigation of mobile robots. *Journal of Intelligent & Robotic Systems*, 102(2):31, 2021.

OmniWarsaw: a City-Scale Omnidirectional Dataset

Karol Majek[1[0000−0002−1351−8496]],
Aleksandra Kos[1,2[0000−0001−9726−4472]],
Miłosz Wojciechowski[1],
Janusz Będkowski[1,3[0000−0003−2630−1947]]

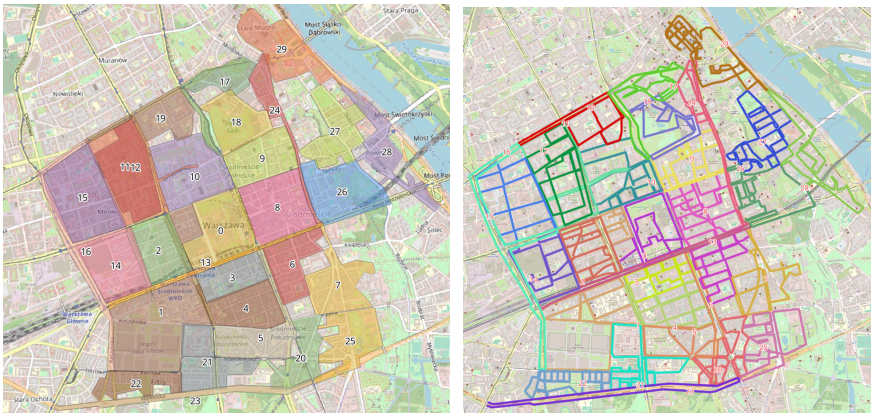
[1]*Cufix, 05-825 Grodzisk Mazowiecki, Poland*
[2]*Poznan University of Technology, 60-965 Poznań, Poland*
[3] *Institute of Fundamental Technological Research, Polish Academy of Science, 02-106 Warsaw, Poland*

Abstract. *In this work we introduce a novel, city-scale, omnidirectional SLAM dataset from a hand-held device. Dataset were collected using open-source open-hardware scanner consisting of non-repetitive scanning pattern Lidar (Livox Mid360) with integrated Inertial Measurement Unit (IMU), GNSS receiver, and a GoPro MAX omnidirectional camera. In total we recorded 188.73 km during in 30 separate session days. We provide raw lidar point clouds, IMU and GNSS readings, anonymized stitched imagery This paper introduces the dataset and describes the data format. Dataset can be accessed via project repository* `https://github.com/MapsHD/OmniWarsawDataset`
Keywords: *omnidirectional camera, object detection, mobile mapping*

1. Introduction and related work

Mobile mapping datasets such as KITTI [1] are used in mobile robotics to improve the Visual and Lidar SLAM algorithms as well as many other tasks including depth estimation and object detection. Since then new devices as well as technology improvement such as non-repetitive scanning patterns in lidar devices e.g. Livox Mid360 community lacks of the new large scale mobile-mapping datasets. However SLAM datasets not related to mobile-mapping using recent hardware exists such as Hilti Dataset [2] which is focused on construction areas.

DOI:10.17388/WUT.2024.0002.MiNI



(a) Covered areas with route ids. (b) Planned scanning paths.

Figure 1: Scanning plan

In a hand held setup new challenges emerge such as less predictable motion model, non-deterministic occlusions. Introduced dataset is currently the largest available dataset using a open-hardware hand-held device in a city-scale environment.

2. OmniWarsaw Dataset

The dataset was collected using open-source open-hardware Mandeye scanner - HDMapping [1] using the provided SLAM methodology [3]. The hardware consists of a LiDAR Livox MID360 with a range 40m @ 10% reflectivity, 70 m @ 80% reflectivity, range precision 1σ: up to 2cm @ 10m, 6-axis IMU ICM40609 with 32kHz sampling.

Data were fused using Normal Distributions Transform [4] and motion model, which acts as a stabilizer and regularizer. Currently only IMU, GNSS, and Lidar data are used for stitching. Total length is 188.73 km, shortest trajectory (route 11 in Figure 1) is 4.85km, and the longest (route 1) is 9.46 km. Data where collected in Warsaw between 13.09.2023-06.11.2023. Figure 2 shows the scanning plan, raw GNSS, panorama image and the final registered point cloud. As the part of the dataset we also provide aerial scanning point clouds acquired as a part of ISOK

[1] `https://github.com/MapsHD/HDMapping`

(Informatyczny System Osłony Kraju) project. Qualitative comparison to ISOK data is shown in Figure 3.

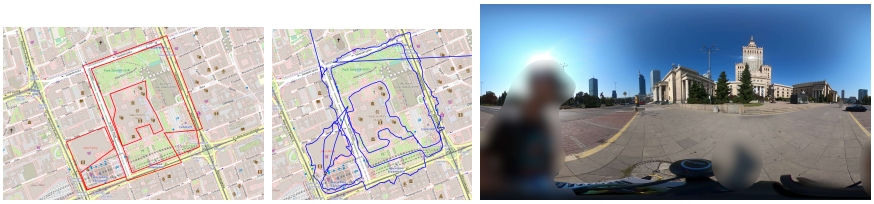
Each of routes has a separate directory with following structure:

```
continousScanning_XXXX
├── GoPro
│   ├── GSXXXXXX_frames
│   │   └── jpg stitched panoramic imagery
│   ├── GSXXXXXX_GPS_Frames.csv frames synced with telemetry
│   └── GSXXXXXX_telemetry.csv telemetry data
├── preview
│   └── lidarYYYY.laz point clouds
├── gnssYYYY.gnss GNSS data
├── imuYYYY.csv odometry data
├── scan_lio_Y.laz point clouds
├── lio_initial_poses.reg with initial poses
├── poses.reg poses after registration
├── session.json metadata file
└── trajectory_lio_Y.csv trajectory
```

Where `continousScanning_XXXX` - is route directory, `XXXX` is route number e.g. 0001. In GoPro directory panoramic imagery, extracted frames, telemetry and GNSS are stored. `GSXXXXXX_frames` directory - contains all extracted frames, in jpg format with 3 seconds interval between every frame All scans except for the route 0 have `GSXXXXXX_GPS_6_Frames.csv` file. File includes following columns: index, date, timestamp in milliseconds, latitude in degrees in WGS84, longitude in degrees in WGS84, altitude in meters, cts - timestamp in milliseconds, image path. `GSXXXXXX_telemetry.csv` is a csv file containing telemetry data. XXXXXX stands for video number, given by the GoPro camera e.g. GS010102. File includes following columns: date, timestamp in milliseconds, latitude in degrees in WGS84, longitude in degrees in WGS84, altitude in meters.

3. Conclusions and Future Work

The new dataset with 360° lidar and imagery is released with the ground truth from ISOK. We provide downloadable lidar part of the dataset while imagery is



(a) Scanning plan (b) Raw GNSS (c) Sample 360° equirectangular image

(d) Aligned scan

Figure 2: Route 0 - visualization

(a) ISOK ground truth point cloud; No walls due to aerial acquisition

(b) Aligned OmniWarsaw dataset visualized in the same projection as ISOK

Figure 3: Qualitative comparison of OmniWarsaw Dataset with ISOK ground truth data

being in the anonymization process[2]. As a feature work we consider releasing a quantitative benchmark of SLAM methods visual and/or lidar as well as benchmark on tiny object detection and tracking and the release of imagery.

Acknowledgment

The research was supported by the Ministry of Education and Science as part of the "Doktorat Wdrożeniowy" program (DWD/5/0203/2021).

References

[1] Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[2] Helmberger, M., Morin, K., Berner, B., Kumar, N., Cioffi, G., and Scaramuzza, D. The hilti slam challenge dataset. *IEEE Robotics and Automation Letters*, 7(3):7518–7525, 2022.

[3] Będkowski, J. Open source, open hardware hand-held mobile mapping system for large scale surveys. *SoftwareX*, 25:101618, 2024. ISSN 2352-

[2]Dataset can be downloaded from `https://github.com/MapsHD/OmniWarsawDataset`

7110. doi:https://doi.org/10.1016/j.softx.2023.101618. URL https://www.sciencedirect.com/science/article/pii/S235271102300314X.

[4] Biber, P. and Straßer, W. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

Positional Encoding for Robot Neural Self-Collision Checking

Bartłomiej Kulecki[1][0000−0002−2820−8212], **Dominik Belter**[1][0000−0003−3002−9747]

[1] *Poznan University of Technology*
60-965 Poznań, Poland
bartlomiej.kulecki@put.poznan.pl

Abstract. *Multiple problems in robotics that are solved using relatively simple multilayer perceptron are defined in low-dimensional feature space. In this paper, we use a multilayer perceptron for self-collision detection of a mobile-manipulating robot. The input vector to the neural network is the configuration of the robot that consists of only six values. We show that enhancing the input vector by features obtained from positional encoding widely used in computer graphics improves classification accuracy. We show the results that suggest that positional encoding improves learning high-frequency functions and better represents higher-frequency details of the trained relation.*
Keywords: *multilayer perceptron, positional encoding, collision checking*

1. Introduction

Despite the development of new efficient neural network architectures, a multilayer perceptron [1] regained attention in robotics and computer vision. The model was successfully applied to create a 3D model of the objects using signed distance transform [2], create the neural model of the scene (NeRF) and render images for new camera poses [3], in Simultaneous Localization and Mapping (SLAM) [4], and control of manipulating robots [5]. The popularity of classical fully connected neural networks comes from the fact that they can be quickly trained to efficiently solve multiple regression and classification problems in robotics.

Multilayer perceptrons applied in robotics take a low-dimensional vector on the input to compute the desired output value. The input vector may represent the

DOI:10.17388/WUT.2024.0002.MiNI

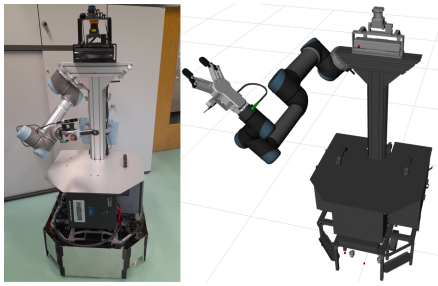


Figure 1. Robot40 and the CAD model of the robot used for implementation of the neural self-collision checking method.

configuration of the robot [6], initial and goal state of the robot [5] or position and view direction in the 3D space [3]. This situation differs from typical problems in computer vision when the input is multi-dimensional and the main challenge is feature extraction. In contrast, this paper focuses on extending the input vector with new features to improve the performance of the neural network. We utilize the neural network to detect self-collisions for a mobile manipulating robot [6] presented in Fig. 1. We focus on features mapping named positional encoding proposed in [3]. The goal of this research is to check whether the positional encoding used in computer vision and graphics can be also used in robotics to improve the results of neural networks.

2. Related Work

The positional encoding proposed in NeRF [3] relies on a combination of trigonometric functions to enhance the input features vector. This approach improves the capability of the neural network to store high-frequency features and preserve the detail of the scene. Much faster than the original version of NeRF is Instant NGP [7]. It utilizes multiresolution hash encoding instead of trigonometric functions to speed up computations on the GPU and reduce the size of the neural network. A review of the Fourier features applied to multiple tasks in computer graphics is presented in [8]. The results suggest that Fourier features improve the performance of MLPs for low-dimensional regression tasks. In this paper, we verify if the same strategy can be applied to self-collision checking – tasks typical in robotics and motion planning. The review of the machine learning approaches for

self-collision checking is presented in [6], the recent methods also use a mixture of Gaussians to infer about collisions [9].

3. Multilayer Perceptron for Collision Checking

In this work, we use a multilayer perceptron with binary output that is trained to detect self-collisions. The training dataset is obtained by randomly sampling the configuration space of the manipulator. We use 200,000 samples consisting of 6 values of joint angles and a collision label (binary output). We divide the dataset into training, testing, and validation sets using a 70/20/10 split. Our baseline approach is using the robot's configuration as an input to a neural network [6]. Each joint θ takes a real value from the range $-\pi$ to π. In this paper, we propose to extend the input to the model using positional encoding [8]. In this approach, a single joint angle value θ is encoded with the function:

$$\gamma(\theta) = (\theta, \sin(2^0\pi\theta), \cos(2^0\pi\theta), ..., \sin(2^{L-1}\pi\theta), \cos(2^{L-1}\pi\theta)) \quad (1)$$

The L parameter determines the length of the MLP input vector which equals $6 + 12 \cdot L$, and $L = 0$ means the input without positional encoding. The considered neural network architectures were obtained by optimizing hyperparameters (using the auto-sklearn framework) for different types of inputs. The obtained models typically consist of three hidden layers and differ in parameters like the number of neurons (from 25 to 176), the learning rate, and the activation function (hyperbolic tangent or ReLU). In addition, we used the original NeRF implementation [3] as well as our implementation based on the information from the paper [3]. These model types have 9 or 10 layers of 256 or 128 neurons.

4. Results

In Tab. 1, we show the results of training the model to classify the robot collision state using the configuration of the robot on the input (without positional encoding) and for the input enhanced with positional encoding. The results include different metrics: accuracy, false negative rate (FNR), and false positive rate (FPR). We tested various values for the L, and in Tab. 1, we show the best results only. We tested the MLP architectures optimized for different values of $L = \{0, 1, 6\}$, original NeRF architecture [3] and our implementation of NeRF architecture (NeRF_{MLP}). The results for MLP(L=1), MLP(L=6), and NeRF_{MLP}

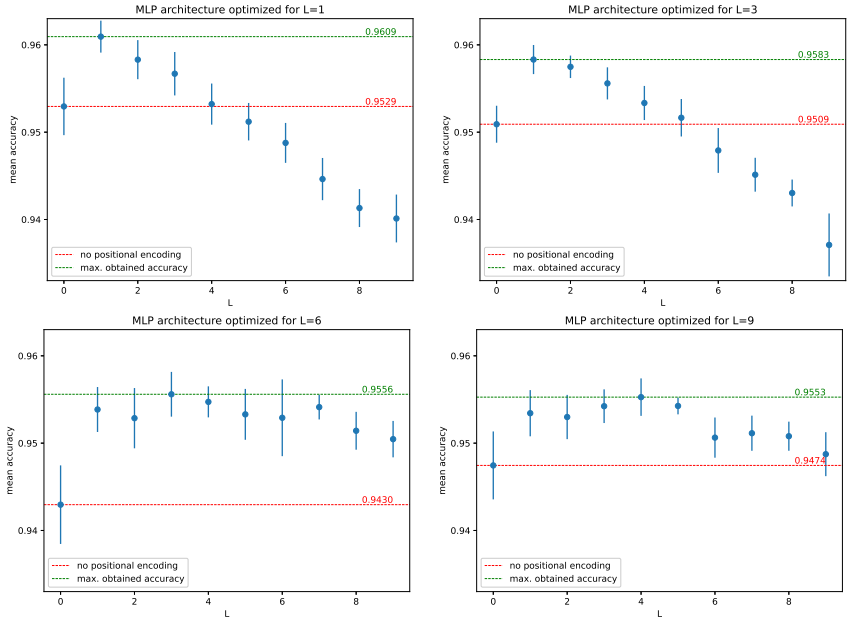


Figure 2. Mean classification accuracy for various positional encoding L parameter values obtained for selected MLP architectures.

show the positive impact of positional encoding – the improvement for them is around 1%. The other two models obtain very similar accuracy for both types of inputs. The results for MLP(L=0) show that the architecture optimized for the raw joint values on the input does not benefit from positional encoding. All architectures obtain an accuracy higher than 94% for any input type. The statistical analysis (two-sample t-test for independent samples) shows that these results are statistically significant.

For selected architectures, we have performed a series of 10 training and testing experiments for each value of L from the range 0 to 9. The obtained mean accuracy values are presented in Fig. 2. As a reference, we show the testing accuracy for $L = 0$ (red line - results without positional encoding) and the best results with positional encoding (green line). The results show that positional encoding improves the accuracy of each model. However, further increasing the L value decreases the accuracy. It means that the relatively simple neural network can not

Table 1. Best classification metrics obtained with and without positional encoding for selected architectures of MLP (statistically significant results underlined).

architecture	MLP(L=0)		MLP(L=1)		MLP(L=6)		NeRF [3]		NeRF_{MLP}	
pos. enc.	off	on	off	on	off	on	off	on	off	on
accuracy [%]	**95.76**	95.70	95.29	**96.09**	94.30	**95.56**	95.81	**95.86**	94.61	**95.59**
FNR [%]	4.78	**4.54**	5.15	**4.91**	5.70	**5.07**	4.43	**4.21**	**5.38**	5.45
FPR [%]	**3.73**	4.09	4.23	**3.07**	5.71	**3.85**	**3.96**	4.09	5.41	**3.93**

extract useful information if the input vector is too large, and for the given problem, the L value should be relatively small (smaller than 5).

5. Conclusions and Future Work

Multilayer perceptrons (MLPs) encounter challenges in learning high-frequency functions, a phenomenon commonly named "spectral bias" [8]. In the case of collision checking, the MLP has problems representing the high-frequency details of the robot's collision workspace. In this work, we show that enhancing features vector for the collision checking neural network using positional encoding improves the classification results. We demonstrate the results for various architectures of MLP and NeRF-based architectures. Most of the neural network architectures take advantage of the additional features and improve the classification results.

In the future, we plan to verify if the proposed strategy works for other common problems in robotics. We will also check if it improves the results of the other classical machine-learning classification methods.

Acknowledgment

The work was supported by the National Science Centre, Poland, under research project no UMO-2019/35/D/ST6/03959 and Poznan University of Technology project no 0214/SBAD/0248.

References

[1] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[2] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174. 2019. doi:10.1109/CVPR.2019.00025.

[3] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 405–421. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58452-8.

[4] Rosinol, A., Leonard, J. J., and Carlone, L. NeRF-SLAM: Real-time dense monocular slam with neural radiance fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3437–3444. 2023. doi:10.1109/IROS55552.2023.10341922.

[5] Kicki, P., Liu, P., Tateo, D., Bou-Ammar, H., Walas, K., Skrzypczyński, P., and Peters, J. Fast kinodynamic planning on the constraint manifold with deep neural networks. *IEEE Transactions on Robotics*, 40:277–297, 2024.

[6] Krawczyk, A., Marciniak, J., and Belter, D. Comparison of machine learning techniques for self-collisions checking of manipulating robots. In *2023 27th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 472–477. 2023. doi:10.1109/MMAR58394.2023.10242571.

[7] Müller, T., Evans, A., Schied, C., and Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. doi:10.1145/3528223.3530127.

[8] Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Curran Associates Inc., Red Hook, NY, USA, 2020. ISBN 9781713829546.

[9] Das, N. and Yip, M. Learning-based proxy collision detection for robot motion planning applications. *IEEE Transactions on Robotics*, 36(4):1096–1114, 2020. doi:10.1109/TRO.2020.2974094.

Task scheduling for autonomous vehicles in the Martian environment

Wojciech Burzyński[1][0009−0003−8668−7754],
Mariusz Kaleta[2][0000−0002−2225−8956]

[1]*Warsaw University of Technology*
Doctoral School
Pl. Politechniki 1, 00-661 Warsaw, Poland
wojciech.burzynski.dokt@pw.edu.pl
[2]*Warsaw University of Technology*
Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
mariusz.kaleta@pw.edu.pl

Abstract. *In the paper, we introduced a novel variant of Electric VRP/TSP, the Solar Powered Rover Routing Problem (SPRRP), to tackle the routing of energy-constrained autonomous electric vehicles for Martian missions. We proposed a basic formulation of the problem based on the graph model that decomposes each Point of Interest into movement, charging, and research tasks. We have also outlined further possibilities for extending the problem.*
Keywords: *task scheduling, electric VRP/TSP, autonomous vehicles, mars exploration*

1. Introduction

A Mars exploration mission is a complex undertaking that requires detailed planning, integration of the newest technologies, and proper budget management. In general, related decisions that have to be made can be divided into strategic, tactical, and operational kinds. The strategic level concerns aim, definition, budget preparation, fleet composition, and strategic resource planning. The tactical level encompasses all decisions related to choosing and planning research tasks to be done within a certain horizon (e.g., weeks or months). At the operational level,

DOI:10.17388/WUT.2024.0002.MiNI

tactical plans are put into action. However, due to high uncertainty and burdens in communication, exploration of extraterrestrial planets requires the use of highly autonomous vehicles. Even though the only mission so far that used multiple vehicles cooperating was *Mars 2020* with rover *Perseverance* and helicopter *Ingenuity* serving as a scout, authors believe that future missions will require cooperation and scheduling of multiple autonomous vehicles. In this paper, we focus solely on the tactical level, assuming that strategic decisions constitute an input to tactical decision models, while the results of tactical models become inputs or may be used to update the aims and tasks of autonomous vehicles at the operational level.

Most of the literature discussing Mars exploration focuses on the operational level, including modeling of kino-dynamic properties of rovers, mechanical properties of Martian soil, computer vision, and reactive trajectory planning. In [1], authors propose a way to evaluate terrain based on satellite imaging and integrate it into a trajectory planner. The problem of maximizing scientific data acquisition by robots in space exploration missions is considered in [2]. The authors proposed a human-in-the-loop multi-agent learning system in which a human team on the Earth identifies a set of Points of Interest (PoIs) and delegates tasks to autonomous agents. In [3], tasks are assigned to robots in a swarm by scientists, and each robot applies its policy to perform its tasks on the simulated Mars surface. The control architecture for multiple-robot planetary outposts (CAMPOUT) has been proposed in [4]. This framework is devoted to cooperating with multiple robots performing tightly coordinated tasks in unpredictable terrain.

In the literature, the scheduling and routing of vehicles were described in terms of the Travelling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP). Their objective is to visit a given set of customers with a vehicle or a fleet of vehicles while satisfying various constraints. In the case of the Martian environment, vehicles visit interesting points to perform research tasks such as soil analysis using a spectrometer. For a comprehensive overview of VRP literature, the reader is referred to [5], and for VRP variants concerning electric vehicles, to [6].

This paper proposes a novel variant of Electric VRP/TSP, that is, the Solar Powered Rover Routing Problem (**SPRRP**), to tackle the routing of energy-constrained electric vehicles that are powered by solar panels such as Martian rovers. Contrary to existing variants, vehicles do not have access to charging stations, and their energy consumption depends on what the vehicle is doing and the time of *sol* (martian day). According to the author's best knowledge, such a variant has not yet been discussed in the literature.

2. Problem and model definition

2.1. Solar Powered Rover Routing Problem (SPRRP)

The task is to plan routes for autonomous vehicles visiting Points of Interest (PoIs); however, some PoIs may remain not visited. Let $Z = \{M, R, C\}$ be the set of task types, where M, R, C means *Movement*, carrying out the *Research*, and *Charging*, respectively. A vehicle travels between PoIs (M), performs the research (R), and optionally can charge (C) at each PoI. We developed a graph model presented in Fig. 1, in which each PoI is modeled with three nodes that allow research and charging to be done in any order at PoI. We assume that all PoIs are connected. In other words, for each PoI, there are edges representing *Movement* task leading to an entry node of every other PoI. All vehicles start at node 0. In the graph model $G = (V, E)$, a task is represented by edge $(i, j) \in E$ and is of type $z_{ij} \in Z$, and node $v \in V$ represents an event related to starting or ending tasks. Each task must be assigned to the vehicle $k \in K$ and is described by starting time t_{ij}^k and duration τ_{ij}^k (we assume constant charging time).

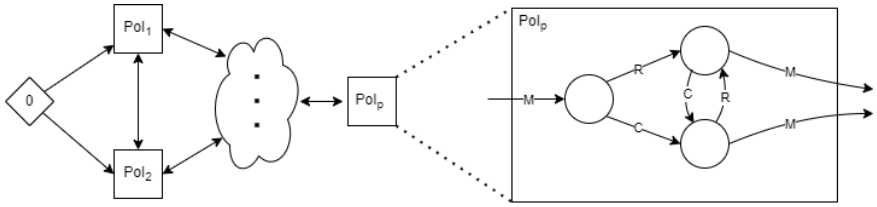


Figure 1. The graph model of the problem. The left part is a flow network representing PoIs with starting node 0, and the right part shows a decomposition of PoI p into specific tasks.

The SPRRP problem can be formulated as a non-linear programming model as follows:

$$\max \sum_{(i,j)\in E} \sum_{k\in K} c_{ij} x_{ij}^k \tag{1}$$

subject to

$$\sum_{i \in V} x_{ij}^k = y_j^k \qquad \forall j \in V \backslash \{0\}, \forall k \in K \tag{2}$$

$$\sum_{i \in V} x_{ji}^k = y_j^k \qquad \forall j \in V, \forall k \in K \tag{3}$$

$$\sum_{k \in K} y_i^k \leq 1 \qquad \forall i \in V \backslash \{0\} \tag{4}$$

$$b_j^k \leq \sum_{i \in V} x_{ij}^k (b_i^k + \Delta E(t_i^k, \tau_{ij}^k, z_{ij})) \qquad \forall j \in V \backslash \{0\}, \forall k \in K \tag{5}$$

$$0 \leq b_i^k \leq B \qquad \forall i \in V, \forall k \in K \tag{6}$$

$$t_0^k = T_0 \qquad \forall k \in K \tag{7}$$

$$t_j^k = \sum_{i \in V} x_{ij}^k (t_i^k + \tau_{ij}) \qquad \forall j \in V \backslash \{0\}, \forall k \in K \tag{8}$$

$$0 \leq t_j^k \leq T_{max} \qquad \forall k \in K, \forall j \in V \tag{9}$$

$$y_j^k \in \{0, 1\}, x_{ij}^k \in \{0, 1\} \qquad \forall k \in K, \forall i, j \in V \tag{10}$$

where x_{ij}^k is a binary variable that equals to 1 if vehicle k traverses edge (i, j), binary variable y_j^k is 1 if vehicle k visits PoI j, b_i^k is battery energy of vehicle k at event i, and t_i^k is the time of occurrence of the event i and vehicle k.

The objective (1) maximizes benefits from completing tasks, where c_{ij} is the benefit assigned to task (i, j). Constraints (2)-(3) ensure that if a vehicle is assigned to event j it must enter and leave node j. Constraint (4) ensures that each node is visited at most once. Note that this assumption does not exclude the case in which a vehicle can physically pass through the PoI previously visited; however, in the graph, there will be a direct connection between the origin and destination. Tracking the battery level b_j^k is done by constraints (5)-(6), where B is battery capacity and b_0^k is initial battery energy. Constraints (7)-(9) track the time of event j, where T_0 and T_{max} are the initial moment and the time limit on the mission, respectively.

Due to constraints (2) and (3) (path consistency) and (8) (subtour elimination), a vehicle must reach any PoI along some path to be able to complete research task at that PoI, and only completion of research tasks increases the objective. Moreover, constraints (5) and (6) force a vehicle to traverse an edge equivalent to pure charging if, otherwise, it cannot perform research tasks due to lack of power.

2.2. Energy model

Let $P(z,t) = P_+(z,t) + P_-(z,t), z \in Z, t \in \mathbb{R}$ be the total power of the vehicle, where z is a type of task performed by a vehicle at time t. $P_-(z,t)$ is the power consumed due to z-type task realization and $P_+(z,t)$ is power gain from solar panels. The energy cost of a task type z starting at t and lasting τ is defined as follows:

$$\Delta E(t,\tau,z) = \int_t^{t+\tau} P(z,t)\mathrm{d}t \tag{11}$$

To simplify the notation and without loss of the generality, we assume simplified power functions as follows:

$$P_-(z,t) = \begin{cases} -a & z = M \\ -b & z = R \\ -c & z = C \end{cases}, \tag{12}$$

$$P_+(z,t) = \frac{sgn(sin(\frac{2\pi t}{T}))}{2} + \frac{1}{2}, \quad T = 1[\text{sol}], \tag{13}$$

where a, b, c are constant power costs defined for each type of task. Introducing different power costs for each task only requires an additional index in the formula (12). $P_+(z,t)$ is defined as a square wave with a period of one sol. This simple model neglects several aspects, including dependence on time of day, weather, heating-up phenomena, etc. Ignoring the position of a given rover can be justified if distances between PoIs are not high, so the differences in the weather and other factors can be neglected.

3. Numerical example

To check the validity of the model, we prepared a simple test case of one vehicle and two PoIs. The graph model of the problem is presented in Fig. 2. Each task can be performed in one unit of time ($\tau_{ij} = 1$), and Table 1 presents the energy consumption for each task. The value of each task is $c_{ij} = 1$. The definition of $P_+(z,t)$ is presented in Fig 3.

To solve the model (1)-(10) we linearized constraints (5) and (8) by introducing additional variables, including binary variables, to represent $\Delta E(t_i^k, \tau_{ij}^k, z_{ij}))$ as a piecewise function and to model product of variables. We used CPLEX to solve the resulting Mixed Integer Problem (MIP).

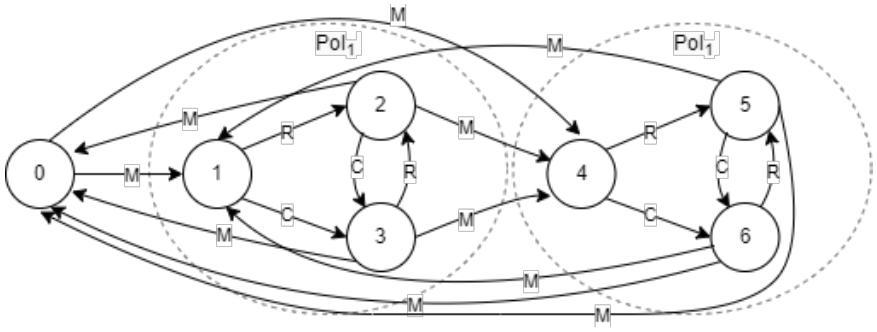


Figure 2. The graph model of the test problem.

Table 1. Energy consumption

task type	**PoI$_1$**	**PoI$_2$**
M Base-PoI	6	10
M PoI-PoI	4	4
R	5	5
C	1	1

To observe the behavior of the optimal results, we manipulated the battery capacity installed on the vehicle. For a battery capacity equal to 4, the optimal solution is to visit only PoI$_1$ and to plan the charging task before performing the research at PoI$_1$. However, when the capacity is increased to 5, the optimal solution still includes only PoI$_1$, but no charging task is planned in that case. With battery capacity ranging from 6 to 10, the model results in visiting both PoIs, but charging task at PoI$_1$ is required. Finally, with a capacity greater or equal to 11, the vehicle is scheduled to serve both PoIs without additional charging tasks. The model behaves correctly and we showed that it can be solved by MIP solvers after linearization. However, since it requires additional variables and constraints, it may suffer from the complexity in case of bigger problems.

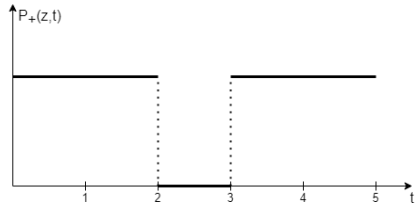


Figure 3. Function of power $P_+(z,t)$ in the numerical example.

4. Problem variants and extensions

Various variants and extensions of SPRRP can be formulated. To grasp the multidimensionality of the problem, we propose the following dimensions describing it:

- vehicles, e.g., capacitated, heterogeneous fleet (specialized vehicles), different energy cost functions;
- tasks/PoIs, e.g., with time windows (for instance, pictures must be taken in specific light conditions), deterministic/stochastic time (task execution and energy budget can be uncertain), obligatory visit of each PoI, complex tasks at PoIs;
- environment, e.g., nondeterministic or dynamic traveling times (best satellite imaging on Mars yields a resolution of 1m/pixel [7] and covers only a small fraction of Mars's surface);
- objective, e.g., distance, time, cost, delays, benefits, risks, including multicriteria versions;

5. Summary

We introduced and formulated a new problem, the Solar Powered Rover Routing Problem, similar to the well-known class of VRP problems but better suited to future Martian missions. Future research should be focused on providing algorithms and numerical results for the introduced model. We also plan to test the model combined with operational algorithms with real vehicles in a simulated

environment on Earth. We also presented possible variants of the problem, demonstrating the plethora of the issue for future research work.

References

[1] Hedrick, G., Ohi, N., and Gu, Y. Terrain-aware path planning and map update for mars sample return mission. *IEEE Robotics and Automation Letters*, 5:5181–5188, 2020.

[2] Colby, M., Yliniemi, L., and Tumer, K. Autonomous multiagent space exploration with high-level human feedback. *Journal of Aerospace Information Systems*, 13(8):301–315, 2016.

[3] Huang, Y., Wu, S., Mu, Z., Long, X., Chu, S., and Zhao, G. A multi-agent reinforcement learning method for swarm robots in space collaborative exploration. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 139–144. 2020.

[4] Huntsberger, T., Pirjanian, P., Trebi-Ollennu, A., Das Nayar, H., Aghazarian, H., Ganino, A., Garrett, M., Joshi, S., and Schenker, P. Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 33(5):550–559, 2003.

[5] Elshaer, R. and Awad, H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering*, 140:106242, 2020. ISSN 0360-8352.

[6] Kucukoglu, I., Dewil, R., and Cattrysse, D. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650, 2021. ISSN 0360-8352.

[7] NASA. Hirise. URL `https://mars.nasa.gov/mro/mission/instruments/hirise/`.

Driving confidence: An uncertainty-aware framework for optimal sensor deployment in autonomous vehicles

Dawid Migacz[1,2] [0000−0001−8607−3443]
Maciej Zięba[2] [0000−0003−4217−7712]
Bartosz Zieliński[1,3] [0000−0002−3063−3621]
Kamil Adamczewski[1] [0000−0002−2917−4392]

[1] *IDEAS NCBR*
name.surname@ideas-ncbr.pl
[2] *Wrocław University of Science and Technology*
name.surname@pwr.edu.pl
[3] *Jagiellonian University in Kraków*
name.surname@uj.edu.pl

Abstract. *Safety and confidence of autonomous systems guided by a range of sensors remains the crucial aspect of their development. We present an uncertainty-aware decision-making framework for optimising sensor usage in autonomous systems, specifically focusing on the choice between LIDAR and RGB cameras. We show that uncertainty-based training improves the object detection task. Moreover, we present uncertainty as a guiding metric to select the most suitable modality for challenging samples. Experiments demonstrate the effectiveness of the proposed framework in enhancing energy efficiency in computationally-contrasting environments.*
Keywords: *decision-making framework, sensor optimisation, modality-switching strategy, uncertainty estimation, energy efficiency, convolutional neural networks, Monte Carlo dropout*

1. Introduction

LIDAR (Light Detection and Ranging) and RGB cameras are two types of sensors commonly used in various applications, including autonomous driving,

DOI:10.17388/WUT.2024.0002.MiNI

robotics, and environmental monitoring. Each technology, with its unique strengths and weaknesses, is tailored for specific applications, particularly in the context of autonomous driving systems.

The RGB cameras, which capture colours and textures, play an important role in identifying traffic signals, road signs, lane markers, and other vehicles. These high-resolution visual data are essential for object recognition and contextual understanding. However, LIDAR excels at measuring distances and operating effectively in low light conditions or in challenging weather, making it indispensable for certain scenarios. The most sophisticated autonomous vehicles employ a combination of sensors, including both LIDAR and RGB cameras. This multi-sensor approach capitalises on the advantages of each technology, enhancing perception accuracy and improving safety.

Given the computational limitations of autonomous systems, there might be situations where prioritising one sensor over another is necessary to minimise computational demands. In such cases, the decision on whether to prioritise LIDAR or RGB cameras hinges on specific requirements.

We explore an uncertainty-based scenario to guide this decision-making process. We consider the importance of uncertainty, a critical factor in high-stakes applications like autonomous driving. Lower uncertainty means more reliable decisions can be made, emphasising the need for precise and reliable sensor data. The decisions are made through a network capable of predicting sensor uncertainty. The uncertainty is used to decide on which part of the network (RGB or LIDAR) we are using, routing the input to the more beneficial one. As inference time is the most important concern, the decision is taken once per image. The uncertainty is calculated using a modification of Monte Carlo Dropout, better suited for convolutional networks, namely using DropBlock instead of dropout.

To sum up, firstly, we notice the importance of uncertainty-aware training and incorporate Dropblock thus improving the baseline object detection accuracy. Secondly, we build router-based uncertainty-aware framework that assesses the confidence of autonomous driving systems and, given computational constraints, allows to select a sensor with the highest confidence. The experimental results show the ability to select appropriate scenes that enhance model performance and optimise its computational resources.

2. Preliminaries

2.1. Object detection

We base our work on CenterNet [1], which is an object detection network that describes the detecting objects as a triplet of key points (two corners and a central keypoint). This approach eliminates the necessity for anchors and the resource-intensive Non-Maximum Suppression (NMS). This change allows the network to output a regressed map of possible central points of objects, avoiding the discontinuity inherent in the traditional approach.

2.2. Uncertainty

It is known that logit values do not provide a reliable measure of uncertainty [2]. Various methods have been proposed to estimate uncertainty, including deep ensembling [3] and Monte Carlo dropout [4]. Deep ensembling, requiring multiple runs of randomly initialised full models, is computationally demanding. Hence, we opt for uncertainty estimation using Monte Carlo Dropout, with the caveat that we employ DropBlock [5] due to its high performance in convolutional neural networks.

3. Method

We train two identical networks separately: one using LIDAR flattened images as input and the other using RGB images. Both networks incorporate DropBlock, which remains enabled during inference, rendering the networks nondeterministic. Subsequently, we compare the inferred central point maps, calculating the variance pixel by pixel. The obtained variances are then averaged to determine the uncertainty of prediction for each modality.

We perform inference on this non-deterministic network multiple times. We denote the output of the ith inference for image number n with modality $\mu \in \{\text{RGB, LIDAR}\}$ by $O^{\mu}_{n,i}$. The network output $O^{\mu}_{n,i}$ is a set that consists of H heads (centre point probability heatmap, bounding box offset, bounding box size, object heading, depth) and every head h is a set of M_h maps (central points probabilities of different classes, individual dimensions of bounding boxes, etc.). Every map is a set of pixels of the same dimensions as the original input, namely $P = 320 \times 96$.

We then denote the output at an individual pixel of a map by $O^{\mu}_{n,i,h,m,\langle x,y\rangle}$. The variance at the point $\langle x, y\rangle$ is then defined as:

$$\mathcal{V}^{\mu}_{n,h,m,\langle x,y\rangle} = \mathrm{Var}_i\left(O^{\mu}_{n,i,h,m,\langle x,y\rangle}\right)$$

and the uncertainty $\mathcal{U}$ of an individual image i with modality m is defined as:

$$\mathcal{U}^{\mu}_{n} = \frac{1}{H}\sum_{h\in O^{\mu}_{n,0}}\frac{1}{M_h}\sum_{c\in O^{\mu}_{n,0,h}}\frac{1}{P}\sum_{\langle x,y\rangle\in O^{\mu}_{n,0,h,m}}\mathcal{V}^{\mu}_{n,h,m,\langle x,y\rangle}.$$

3.1. Modality switching

Our aim is to optimise the energy usage of an edge device (such as a drone or autonomous vehicle). We assume that the RGB camera is always enabled as it is necessary for the operator controls. We selectively activate the LIDAR sensor on the basis of necessity, specifically to handle the most challenging samples. Our strategy involves identifying samples where the difference $\mathcal{U}^{\mathrm{LIDAR}}_{n} - \mathcal{U}^{\mathrm{RGB}}_{n}$ is maximised and processing them using the LIDAR network. However, since this difference is unavailable at inference time (calculating it directly would incur prohibitive costs outweighing any potential savings), we estimate it using a shallow convolutional neural network, taking an RGB image as input.

4. Experiments

We conduct experiments using the KITTI dataset [6] and a single NVIDIA A100 GPU. The training process involves LIDAR and RGB networks, each trained for 300 epochs with DLA34 as the backbone [7]. Additionally, the uncertainty estimation network, featuring two hidden layers, undergoes training for 40 epochs.

4.1. Dropblock-based object detection

First, we show that incorporating Dropblock-based training improves the baseline Centernet network. We report average precision ($\mathrm{AP}|_{R_{40}}$, see [8]) at the intersection over union, IoU= 0.7 where scale $\in$ {small, medium, large}, denotes the scale of object. We use Dropblock= 0.15 which provides the best results.

	small	medium	hard
RGB Pretrained	17.3234	13.9032	11.9186
RGB Dropblock	17.9642	14.1806	12.0896
LIDAR Pretrained	58.3248	42.8515	37.1175
LIDAR Dropblock	**61.0089**	**46.4143**	**40.4691**

Table 1. Comparison of object detection performance using RGB and LIDAR with and without DropBlock.

4.2. Modality switching

We calculate the $AP|_{R_{40}}$ of 3D bounding box predictions at IoU = 0.7, where some of the images are processed using LIDAR input and the remaining ones with RGB image. In Fig. 1 we present the results. The percentage of input data that uses LIDAR network ranges from $\{1, \ldots, 30\}$). The fraction of these images is selected via three approaches, random selection, ground-truth uncertainty, and estimated uncertainty. For clarity we present results for large objects, however they are similar for the other classes.

Notably, estimated uncertainty outperforms the ground truth. We hypothesize this is due to using a shallow prediction network that forms a regularising effect, effectively denoising the true uncertainty. The secondary advantage is that the low computational cost of the router network incurs minimal energetic overhead during inference, leading to practical savings.

As a metric, we use the area under the curve (AUC) bounded with upper and lower precision. Please note that in this context, AUC is used literally and is unrelated to precision and recall.

method	random	true uncertainty	estimated uncertainty
AUC	0.423	0.458	**0.528**

Table 2. Comparison of Area Under the Curve (AUC) for different uncertainty estimation methods.

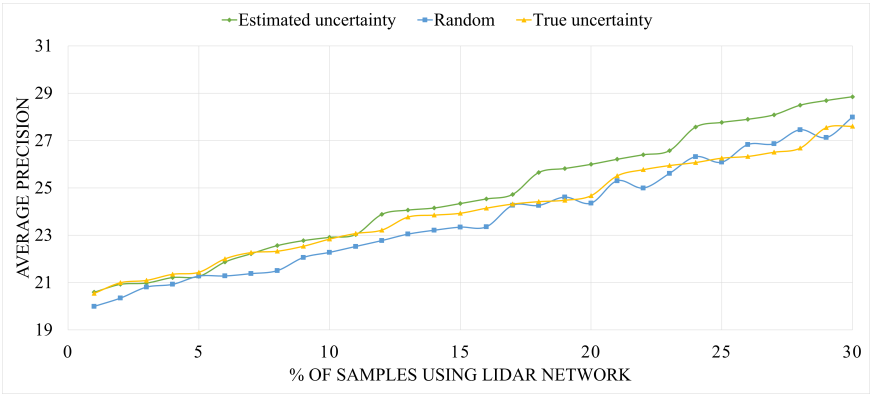


Figure 1. Comparison of the success of various uncertainty estimation methods in modality switching

4.3. Uncertainty visualisation

Fig. 2 illustrates predictions and uncertainties from both RGB and LIDAR networks. The top-left and top-right corners show RGB and LIDAR images with respective bounding-box predictions. In the middle, probability maps indicate the likelihood of specific points being the centre of a car's bounding box for both networks. The bottom part displays pixel-wise uncertainty maps for RGB and LIDAR predictions, offering a visual understanding of the models' confidence across the images.

5. Discussion

In this work, we presented a framework aimed at optimizing sensor deployment in autonomous vehicles by leveraging an uncertainty-aware decision-making process. To enhance object detection, the framework utilizes uncertainty to guide the choice of sensor modality for challenging samples, thereby improving energy efficiency in computationally diverse environments. The current set-up uses standard and widely applicable architectures with a range of possible improvements that include 3D-convolution network. Moreover, this framework is fully transferable to approaches other than CenterNet, defining uncertainty as variance of the

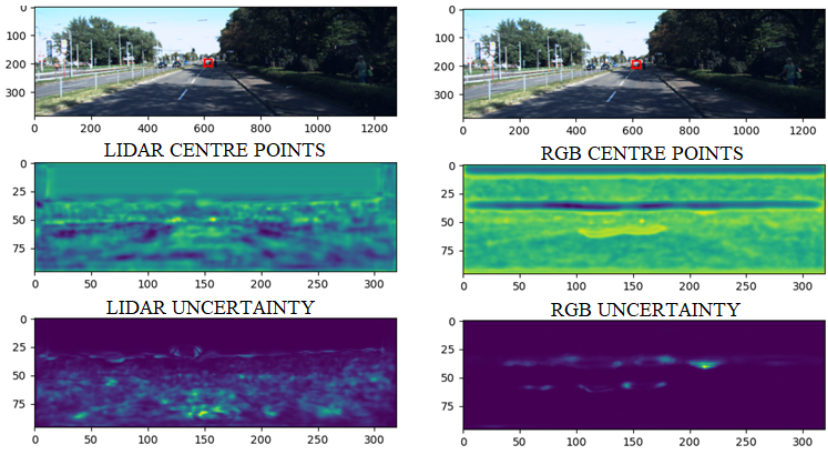


Figure 2. Centre point maps (brighter = higher probability) and uncertainty maps (brighter = higher uncertainty)

second-to-last layer. However, if the architecture of the network differs, the technique becomes less explainable. While our framework targets sensor selection for entire scenes, oftentimes both sensors can be utilized. Such sensor availability leads to possible fusion approaches that may, for example, focus to reduce region uncertainty thus further increasing our confidence. Lastly, the proposed method has been used for object detection but can readily be applied for other tasks such as scene segmentation or image classification.

Acknowledgement

We gratefully acknowledge the Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016814.

References

[1] Zhou, X., Wang, D., and Krähenbühl, P. Objects as points, 2019.

[2] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[3] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416. Curran Associates Inc., Red Hook, NY, USA, 2017. ISBN 9781510860964.

[4] Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, New York, New York, USA, 2016. URL `https://proceedings.mlr.press/v48/gal16.html`.

[5] Ghiasi, G., Lin, T.-Y., and Le, Q. V. Dropblock: a regularization method for convolutional networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10750–10760. Curran Associates Inc., Red Hook, NY, USA, 2018.

[6] Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[7] Yu, F., Wang, D., and Darrell, T. Deep layer aggregation. *CoRR*, abs/1707.06484, 2017. URL `http://arxiv.org/abs/1707.06484`.

[8] Simonelli, A., Bulo, S. R., Porzi, L., López-Antequera, M., and Kontschieder, P. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999. 2019.

Chapter 9

Problem Solving and Optimisation

Track Chairs:

- prof. Jarosław Arabas - Warsaw University of Technology
- prof. Karol Opara - Systems Research Institute of the Polish Academy of Sciences
- prof. Szymon Łukasik - AGH University of Krakow

Hall of Fame in Coevolutionary Algorithm for Stackelberg Security Games

Adam Żychowski[1][0000−0003−0026−5183],
Jacek Mańdziuk[1][0000−0003−0947−028X],
Andrew Perrault[2][0000−0002−5062−7958]

[1]*Warsaw University of Technology, Faculty of Mathematics and Information Science*
adam.zychowski@pw.edu.pl jacek.mandziuk@pw.edu.pl
[2]*The Ohio State University, Department of Computer Science and Engineering*

Abstract. *Stackelberg Security Games (SSGs) is a popular game-theoretic model for strategic interactions between a Defender and an Attacker. The computational challenges of identifying optimal strategies for larger games led to the development of the CoEvoSG coevolutionary method. This paper introduces an extension to CoEvoSG, incorporating a Hall of Fame (HoF) mechanism with Mixed Nash Equilibrium. The HoF stores successful strategic configurations, enhancing algorithm robustness. Results across distinct game types demonstrate that the proposed method consistently outperforms the baseline CoEvoSG algorithm with standard HoF approach. This improvement is achieved with minimal computation time increase.*
Keywords: *Stackelberg Security Games, Evolutionary Computation*

1. Introduction

Stackelberg Security Games (SSGs) represent a widely applicable game model effectively deployed across various real-world security domains, including security patrolling [1, 2], poaching prevention [3], terrorist deterrence [4], and cybersecurity [5]. The challenge of identifying the optimal pair of players' strategies (Stackelberg Equilibrium) for SSGs is proven to be NP-hard, prompting the development of numerous approximation methods in the literature. This study builds upon the

DOI:10.17388/WUT.2024.0002.MiNI

CoEvoSG coevolutionary method, which has demonstrated superior performance in terms of computation time and the ability to address larger games that were previously intractable. In this paper we extend the CoEvoSG method by the Hall of Fame (HoF), containing strategies of the Nash Equilibrium for the current state of players' populations. This addition enhances the proposed method, yielding improved results with only a marginal increase in computation time.

1.1. Problem definition

SSGs involve two players: the Defender (D) and the Attacker (A). Each game consists of m time steps during which both players simultaneously choose actions. A player's *pure strategy* σ_P ($P \in \{D, A\}$) is a sequence of their actions in consecutive time steps: $\sigma_P = (a_1, a_2, \ldots, a_m)$. The set of all possible pure strategies of player P is denoted by Σ_P. Furthermore, the player's *mixed strategy* is a probability distribution $\pi_P \in \Pi_p$ over Σ_P, where Π_p is the set of all mixed strategies for player P. For any pair of strategies (π_D, π_A) the expected payoffs for the players are denoted by $U_D(\pi_D, \pi_A)$ and $U_A(\pi_D, \pi_A)$, resp. The goal of the game is to find the *Strong Stackelberg Equilibrium* (SSE), i.e. a pair of strategies (π_D, π_A) satisfying the following conditions:

$$\pi_D = \arg\max_{\bar{\pi}_D \in \Pi_D} U_D(\bar{\pi}_D, BR(\bar{\pi}_D)), \qquad BR(\pi_D) = \arg\max_{\pi_A \in \Pi_A} U_A(\pi_D, \pi_A).$$

The first one optimizes the selection of the Defender's strategy π_D under the assumption that the Attacker always opts for the best response strategy ($BR(\pi_D)$) to the Defender's committed strategy. Both players determine their strategies at the beginning of a game (first the Defender and then the Attacker). Once the strategies are chosen, they remain fixed throughout the duration of the game.

2. Proposed method - Mixed Nash Equilibrium Hall of Fame

The CoEvoSG algorithm [6] is a coevolutionary approach designed to address the computational challenges associated with evaluating strategies in SSGs. In standard evaluations, the Attacker's pure strategies are exhaustively explored to determine the best response to the Defender's strategy. This process becomes impractical for larger games or continuous strategy spaces. CoEvoSG maintains two populations, one for the Defender's mixed strategies and the other for the Attacker's pure strategies. The algorithm operates by alternating modifications of

the Attacker's and Defender's populations, and evolving them through a specified number of generations. The Attacker's population represents pure strategies which are developed with evolutionary operators such as crossover and mutation. **Crossover** combines strategies by swapping actions, while **mutation** introduces new actions, both contributing to the exploration of the strategy space.

Evaluation process involves the Defender's population being assessed against a subset of the Attacker's strategies. The best Attacker's response is determined, and the Defender's payoff against this response is used as the fitness value for the evaluated Defender's strategy. The evaluation of the Attacker's population is more complex, considering the adaptability of the best response to different Defender strategies. Namely, in order to ensure the effectiveness of the Attacker's strategies against multiple Defender's strategies (not just the best one) and at the same time to avoid oscillations, $N_{top} = 20$ highest-fitness Defender's strategies are used for each Attacker's strategy evaluation. **Selection** determines which individuals advance to the next generation. It includes elite preservation and binary tournaments. The algorithm concludes when a predefined number of generations is reached or when no improvement in the best-found solution (Defender's payoff) is observed over a specified number of consecutive generations.

A more detailed description of the CoEvoSG algorithm can be found in [6].

2.1. Mixed Nash Equilibrium (MNE) Hall of Fame

We expanded the CoEvoSG algorithm by incorporating the Hall of Fame (HoF) archive mechanism featuring Nash Equilibrium. HoF is a well-established concept in evolutionary computation which typically consists in the inclusion of the best individual in each generation to contribute to subsequent evaluations.

Please note, that HoF differs from the well-known elitism mechanism, as the latter consists in preserving the best individuals between generations (in the selection process), so as not to lose the overall best solution, while HoF extends this concept by storing the best solutions found throughout the entire evolutionary process for their further usage or for reference.

In contrast, our method deviates by integrating the MNE strategy instead of a single best individual. A similar approach was previously proposed in [7] for optimizing robust decision trees.

Two distinct HoFs are maintained, one for the Defender's population and another for the Attacker's population. Following each generation, the MNE strategy is computed based on the current players' populations. The resultant mixture com-

prises the Attacker's pure strategies and the Defender's mixed strategies, which are subsequently appended to the respective HoFs. Another modification to CoEvoSG is introduced in the evaluation procedure. Individuals (strategies) from a given population (Defender's or Attacker's) undergo evaluation not only against individuals from the adversarial population but also with each corresponding element within their respective HoF.

The HoF acts as a repository of historical knowledge containing strategic interactions between the Defender and the Attacker over multiple generations. By storing MNE in the HoF, the algorithm preserves strategic configurations that have demonstrated success in dealing with a variety of opponent strategies. It improves algorithm robustness since strategies in the next generations are evaluated also against effective strategic configurations from the past rather than relying on a single best individual from the current adversarial population.

3. Results

The proposed method has been tested on 2 distinct types of Security Games: Search Games (SEG) [8] and FlipIt Games (FIG) [9]. The same game instances were also used for CoEvoSG evaluation [6]. Please refer to [10] for a detailed description of the rules and characteristics of the games. To maintain consistency and ensure a fair comparison, the baseline CoEvoSG was configured with the same set of parameters as reported in [6].

	C2016	O2UCT	EASG	CoEvoSG	CoEvoSG+HoF	CoEvoSG+NEHoF
5	0.890	0.887	0.886	0.886	0.886	0.887
10	0.854	0.848	0.847	0.845	0.845	0.849
15	0.811	0.805	0.802	0.798	0.801	0.806
20	-	0.779	0.780	0.772	0.775	0.776
25	-	-	0.754	0.746	0.751	0.754
30	-	-	-	0.730	0.732	0.735
40	-	-	-	0.722	0.726	0.733

Table 1. Averaged Defender's payoff with respect to game nodes for FIG.

Tables 1 and 2 present the results obtained from five distinct methods. *C2016* [11] represents an exact method based on Mixed Integer Linear Programming, ensuring the return of optimal Defender's strategies. However, its applicability is constrained by extensive computation times, which makes it suitable only

	C2016	O2UCT	EASG	CoEvoSG	CoEvoSG+HoF	CoEvoSG+NEHoF
15	0.122	0.116	0.115	0.115	0.115	0.116
20	0.117	0.107	0.106	0.101	0.104	0.106
25	-	0.119	0.117	0.115	0.116	0.119
30	-	-	0.136	0.135	0.135	0.135
40	-	-	-	0.150	0.152	0.156
50	-	-	-	0.139	0.144	0.146

Table 2. Averaged Defender's payoff with respect to game nodes for SEG.

for small-scale game instances. O2UCT [12] is a heuristic method which utilizes the Upper Confidence Bounds applied to Trees (UCT) algorithm - a variant of the Monte Carlo Tree Search method (MCTS). *EASG* [10] is an evolutionary algorithm maintaining a single population of Defender's strategies. While utilizing the same evolutionary operators as the CoEvoSG Defender's population, EASG evaluates candidate Defender's strategies against all possible pure Attacker's strategies, ensuring precise payoff calculations. However, this precision results in a significantly longer computation time compared to CoEvoSG. *CoEvoSG+HoF* is a baseline CoEvoSG algorithm augmented with standard HoF - the best individual from the current population is added to the respective HoF after each generation. *CoEvoSG+NEHoF* represents the proposed method incorporating MNE strategies into the respective HoFs.

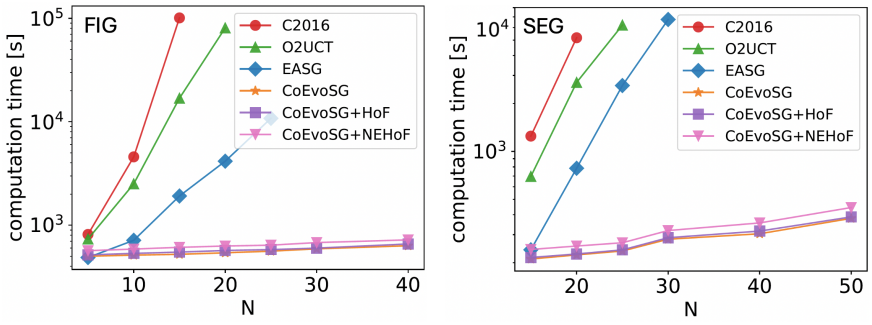


Figure 1. Computation times with respect to game nodes (N) for FIG and SEG.

Results across all datasets consistently indicate that the inclusion of an additional HoF element has a positive impact on the outcomes. In the majority of

games, CoEvoSG+NEHoF achieved higher Defender's payoffs than the baseline CoEvoSG algorithm. Furthermore, it outperforms CoEvoSG+HoF, supporting the claim that utilizing MNE results for HoF, as opposed to a simple best individual, is advantageous. Comparisons with EASG, which is a reference point for the optimal Defender's strategy evaluation, reveal that CoEvoSG+NEHoF achieves competitive results while exhibiting significantly lower computation times.

Figure 1 depicts the averaged computation times for each method. Notably, for all three coevolutionary algorithms, these times are significantly lower than those for other methods. The introduced overhead caused by HoF maintenance and MNE computation is insignificant. On average, CoEvoSG+HoF and CoEvo+NEHoF computation times are respectively 4% and 13% higher than plain CoEvoSG method that does not use the HoF mechanism.

4. Summary

In this study, we introduced an enhanced coevolutionary algorithm designed to address SSGs. It incorporates a HoF mechanism based on MNE derived from players' population strategies. The proposed method outperforms the baseline algorithm (absent of HoF), with only slight increase in computation time.

Acknowledgment

Adam Żychowski was funded by the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

References

[1] Sinha, A., Fang, F., An, B., Kiekintveld, C., and Tambe, M. Stackelberg Security Games: Looking Beyond a Decade of Success. In *27th IJCAI Conference*, pages 5494–5501. 2018.

[2] Karwowski, J. and Mańdziuk, J. A Monte Carlo Tree Search approach to finding efficient patrolling schemes on graphs. *European Journal of Operational Research*, 277:255–268, 2019.

[3] Żychowski, A., Mańdziuk, J., Bondi, E., Venugopal, A., Tambe, M., and Ravindran, B. Evolutionary approach to Security Games with signaling. *31st IJCAI Conference*, pages 620–627, 2022.

[4] Karwowski, J., Mańdziuk, J., Żychowski, A., Grajek, F., and An, B. A memetic approach for sequential security games on a plane with moving targets. In *Proceedings of the 33rd AAAI conference*, volume 33, pages 970–977. 2019.

[5] Mańdziuk, J. and Żychowski, A. Duel-based neuroevolutionary method for stackelberg security games with boundedly rational attacker. *Applied Soft Computing*, 146:110673, 2023.

[6] Żychowski, A. and Mańdziuk, J. Coevolution of players strategies in Security Games. *Journal of Computational Science*, 68:101980, 2023.

[7] Żychowski, A., Perrault, A., and Mańdziuk, J. Coevolutionary Algorithm for Building Robust Decision Trees under Minimax Regret . In *38th AAAI Conference*. 2024.

[8] Bošanský, B. and Čermak, J. Sequence-Form Algorithm for Computing Stackelberg Equilibria in Extensive-Form Games. In *29th AAAI Conference*, pages 805–811. 2015.

[9] Van Dijk, M., Juels, A., Oprea, A., and Rivest, R. L. Flipit: The game of "stealthy takeover". *Journal of Cryptology*, 26(4):655–713, 2013.

[10] Żychowski, A. and Mańdziuk, J. Evolution of Strategies in Sequential Security Games. In *20th AAMAS Conference*, pages 1434–1442. 2021.

[11] Čermák, J., Bošanský, B., Durkota, K., Lisý, V., and Kiekintveld, C. Using correlated strategies for computing stackelberg equilibria in extensive-form games. In *30th AAAI Conference*, pages 439–445. 2016.

[12] Karwowski, J. and Mańdziuk, J. Double-oracle sampling method for Stackelberg Equilibrium approximation in general-sum extensive-form games. In *34th AAAI Conference*, volume 34, pages 2054–2061. 2020.

Enhancing car clip classification accuracy with limited data

Martyna Poziomska[1,2][0000−0002−3754−7515]
Krzysztof Romanowski[1,3][0009−0005−3362−8210]
Konrad Małek[1][0000−0001−7710−3559]

[1]*Łukasiewicz Research Network*
Automotive Industry Institute
Jagiellońska 55, 03-301 Warsaw, Poland
konrad.malek@pimot.lukasiewicz.gov.pl
[2]*University of Warsaw*
Faculty of Physics
Pasteura 5, 02-093 Warsaw, Poland
martyna.poziomska@gmail.com
[3]*Polish-Japanese Academy of Information Technology*
Faculty of Informatics
ul. Koszykowa 86, 02-008 Warsaw, Poland
krzysztof.romanowski@pimot.lukasiewicz.gov.pl

Abstract. *This project addresses the challenge of training neural networks with limited data using a small dataset of car clips. Focusing on the CoAtNet architecture, the objective is to achieve accurate classification despite data scarcity. Two key strategies, data augmentation and pretraining on a synthetic dataset, are employed to enhance the model's generalization capabilities. The real dataset (RD) consists of 164 car clips, each with only 50 images, reflecting real-world conditions. In contrast, the synthetic dataset (SD) comprises 3000 images per clip, providing diverse instances for pretraining. Data augmentation, including rotation and scaling, aims to artificially enrich the RD. The project methodology involves a two-phase neural network pretraining process. First, extensive training on SD forms a robust foundation for subsequent fine-tuning on RD. The impact of data augmentation using the torchvision library is explored, enhancing the model's ability to generalize effectively. Results indicate that training solely on RD*

DOI:10.17388/WUT.2024.0002.MiNI

with augmentation yields an accuracy of 0.844, while exclusive training on SD results in a lower accuracy of 0.347. Notably, pretraining on SD followed by fine-tuning on RD, with augmentation, achieves the highest accuracy of 0.944. The project also investigates the model's sensitivity to varying amounts of real-world data during fine-tuning. Even with reduced percentages, the model exhibits robust adaptability. In conclusion, the combination of data augmentation and synthetic pretraining significantly improves the CoAtNet model's classification accuracy for diverse car clips, demonstrating its potential in scenarios with limited real-world data.
Keywords: *augmentation, synthetic data, pretraining, artificial intelligence*

1. Introduction

In the field of machine learning, the challenge of training neural networks with limited data is particularly pronounced. This project specifically aims to address this issue by utilizing a small dataset of car clips. The primary objective is to train a model, specifically the CoAtNet [1] architecture, for accurate classification of these diverse car clips.

To overcome the data scarcity, two pivotal strategies are implemented: data augmentation and pretraining on a synthetic dataset. Data augmentation involves creating diverse variations of the limited dataset through techniques like rotation and scaling. This artificial expansion aims to enrich the neural network's ability to generalize and make precise predictions when confronted with novel, unseen data.

Furthermore, the neural network undergoes pretraining on an extensive synthetic dataset generated to accurately represent different car clips. This synthetic dataset serves as an invaluable resource for the initial training phase before fine-tuning on the smaller, real-world dataset.

By combining the power of data augmentation and pretraining on a synthetic dataset, this project endeavors to showcase the effectiveness of the CoAtNet model in accurately classifying a wide array of automotive clips, even when faced with the inherent challenges of a limited dataset.

The authors in [2] used a similar method to the one proposed in the project, but for a different purpose. Using 3D models, they diversified the background of the target object, and then, by analyzing the 2.5D image (with depth), they taught the model to detect the target object. This method did not necessarily focus on replicating real conditions, but rather on artificial background noise. Instead, [3] used the fractal generation method to create a synthetic dataset for classification pre-training. Fractals have shapes similar to real objects, but they do not look

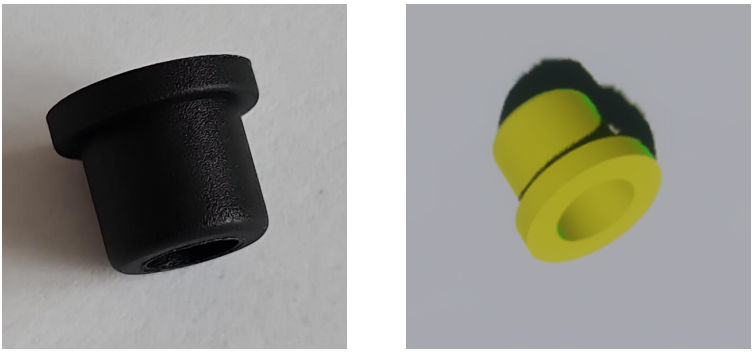
like real photos. The authors of [4] analyzed various approaches to using generative models to create a synthetic dataset for pre-training, but these images are also not able to reproduce reality so well. The solution proposed in the PoC project focused on achieving the most realistic synthetic images possible. Images of objects against a background similar to the target were used in pre-training, and the objects themselves, thanks to appropriately prepared randomization and augmentation, also reproduced the real ones with high accuracy.

2. Data

The real dataset (RD) comprises photographs of 164 distinct car clips, taken using various models of mobile phones. Despite the notably limited data (50 images for each clip), this dataset reflects real-world photographic conditions.

On the other hand, the synthetic dataset (SD) consists of an 3000 images for each clip category. These data were generated using an advanced synthetic data generator with Unity engine, allowing precise modeling and representation of each of the 164 clips. This synthetic dataset is a crucial component for pretraining the model, providing a substantial amount of diverse instances to enhance the overall classification effectiveness.

A sample images from SD and RD are shown in Figure 1.



(a) RD (b) SD

Figure 1: Sample images for a) RD and b) SD

3. Methods

The project methodology encompassed a meticulous exploration of strategies to train a CoAtNet model for the classification of 164 car clips, despite the challenge of limited real-world data. All comparative analyses were conducted using a robust 5-fold cross-validation approach, with accuracy serving as the primary metric for assessing model performance. The chosen metric is an appropriate measure of the predictive power as the RD is balanced. For each training session, the dataset was split into training and testing in an 80/20 ratio.

In the realm of neural network pretraining, a two-phase process was employed. Initially, the model underwent training on a substantial SD, generating a diverse foundation for subsequent fine-tuning on the RD. This approach aimed to leverage the richness of SD to enhance the model's overall learning capacity.

A critical aspect of the study involved investigating the impact of data augmentation using the torchvision library [5]. By introducing diverse transformations such as rotation and scaling, the goal was to artificially expand the RD and boost the model's ability to generalize effectively.

The exploration of pretraining on an SD aimed to discern the model's response to a comprehensive set of diverse instances. This phase provided insights into how well the model could adapt its learned features to real-world scenarios after initial exposure to synthetic data.

Additionally, the study sought to determine the optimal amount of real-world data required for effective fine-tuning. This analysis aimed to identify the threshold at which the model achieved high-performance results, optimizing the utilization of scarce real-world data.

4. Results

The CoAtNet model underwent a thorough evaluation, exploring various training configurations and testing scenarios. Table 1 summarizes the accuracy results for different training scenarios, with subsequent tables (2, 3, 4) providing detailed insights.

The first set of experiments involved training the model solely on real-world data, exploring the impact of data augmentation, as depicted in Table 1.

In the second set of experiments, the model was exclusively trained on synthetic data and evaluated on real-world data, as shown in Table 2.

Table 1: Model Evaluation: Real-Only Training and Testing

Training	Testing	Augmentation	Accuracy
RD	RD	-	0.786 ± 0.008
		RD	0.844 ± 0.017

Table 2: Model Evaluation: Synthetic-Only Training and Real-World Testing

Training	Testing	Augmentation	Accuracy
SD	RD	SD	0.347 ± 0.030
		-	0.023 ± 0.003

Table 3: Model Evaluation: Synthetic Pretraining and Real Fine-Tuning

Pretraining	Fine-Tuning	Testing	Augmentation	Accuracy
SD	RD	RD	-	0.925 ± 0.005
			SD, RD	0.944 ± 0.009

The third set of experiments involved pretraining the model on synthetic data and fine-tuning on real-world data, exploring the impact of augmentation, as detailed in Table 3.

Table 4: Model Evaluation: Real Data Amount for Pretraining

Pretraining	Percentage of RD for Fine-Tuning	Testing	Augmentation	Accuracy
SD	80%	RD	SD, RD	0.944 ± 0.009
	50%			0.916 ± 0.006
	30%			0.863 ± 0.004
	15%			0.760 ± 0.005

Finally, the impact of varying amounts of real-world data for fine-tuning was investigated, as presented in Table 4. Different training percentages were examined, providing insights into the model's sensitivity to the quantity of real-world data during the fine-tuning phase.

5. Conclusions

In addressing the challenge of limited data for training neural networks, this project utilized a small dataset of car clips to train the CoAtNet model. Leveraging data augmentation and pretraining on a synthetic dataset, the project aimed to enhance the model's ability to accurately classify diverse automotive clips.

The evaluation results showcased the impact of different training strategies on model performance. Training solely on real-world data with augmentation yielded an accuracy of 0.844, while exclusive training on synthetic data resulted in a lower accuracy of 0.347. Notably, pretraining on synthetic data followed by fine-tuning on real-world data, with augmentation, demonstrated the highest accuracy of 0.944.

Moreover, the project explored the sensitivity of the model to varying amounts of real-world data during fine-tuning. The model achieved high performance even with a reduced percentage of real-world data, indicating robust adaptability.

In the future, it is worth considering of class-aware AI model, as now the model will classify all possible objects as a car clip.

In summary, the combination of data augmentation and synthetic pretraining significantly improved the CoAtNet model's classification accuracy for a diverse set of car clips, showcasing its potential for effective utilization in scenarios with limited real-world data.

References

[1] Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. 2021.

[2] Planche, B., Wu, Z., Ma, K., Sun, S., Kluckner, S., Lehmann, O., Chen, T., Hutter, A., Zakharov, S., Kosch, H., and Ernst, J. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5d recognition. In *2017 International Conference on 3D Vision (3DV)*, pages 1–10. 2017. doi: 10.1109/3DV.2017.00011.

[3] Kataoka, H., Okayasu, K., Matsumoto, A., Yamagata, E., Yamada, R., Inoue, N., Nakamura, A., and Satoh, Y. Pre-training without natural images, 2021.

[4] He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., and Qi, X. Is synthetic data from generative models ready for image recognition?, 2023.

[5] Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020.

Return and Restart Methods in the Hybrid Approach to Procedural Generation of Dungeons for Computer Games

Mateusz Darul, Dominik Szajerman[0000−0002−4316−5310]

Institute of Information Technology
Lodz University of Technology
al. Politechniki 8, 93-590 Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract. *This article deals with the topic of procedural dungeons generation. The dungeons are a specific type of levels in computer games. A recently proposed hybrid method consisting of context free grammar and cellular automata showed interesting results in generating dungeons. The proposed method extends the base method by two additions. The Return Method, which changes the room selection algorithm where the repeated generation starts when a dead end is reached. And the Restart Method, in turn, allows for generation of more branched dungeons for deeper exploration during the game. Generated dungeons perform better in terms of being "interesting". Moreover, a greater time efficiency of the generator was achieved.*
Keywords: *computer games, procedural content generation, dungeons*

1. Introduction

The subject of procedural content generation in games is becoming more and more popular. Dungeons are a specific type of level in games whose design has its own conventions and limitations. The topic of generators is very interesting and has great potential, because creating computer games is a time-consuming activity, and any way to speed up this project is worth attention.

Procedural generation of dungeons takes into account various algorithms. Cellular automata could be used to generate "interesting" spaces. A good example of their use is the work [1], which presents a cave generator suitable, for example, for

DOI:10.17388/WUT.2024.0002.MiNI

the game *Terraria* or *Dead Cells*. Generative grammar algorithms are often used in generating missions, i.e. the proper order of gameplay elements. In order to be able to generate space, they must be used in connection with other methods, for example with cellular automata, as in [2], or with a shape grammar as in [3] and [4] Most often, the output of this could generate levels for games like *The Legend of Zelda: A Link to the Past* or *The Binding of Isaac*. A hybrid solution [2] is combining context free grammars and cellular automata. The former of the algorithms is used to generate a mission – description of dungeon being generated, the latter processes the data to get the distribution of elements in space. Dungeon consist of a grid of adjacent, separate, and equally sized rooms. Up to four doors in the main directions can lead from each room to its neighbors. A key is required to open some of them.

2. Method

The described hybrid solution [2] is referred as the base method later in this paper. The detailed rules of the generation are as follows. The first one chooses a pseudo-random direction of movement to any of the adjacent positions on the grid. If there is no room there, it is created and its type is generated based on the next symbol from the mission description. The process is repeated until the output data from grammar is exhausted, i.e. the end room is reached. The second rule deals with the dungeon subsections. A subsection is a distinct group of rooms that are delimited by a start room and the first locked door, or by a pair of locked doors, or by the last locked door and the end room. The *average centre point* of positions of rooms in the subsection is calculated. Then the empty place closest to it and simultaneously adjacent to some room from the subsection is chosen as the place for the next room. It allows the subsection to grow as a natural cluster. "Subsection Search on Halt" algorithm proposed by authors uses the first rule until a dead end is encountered i.e. there is no empty cells adjacent to the current one, in which case works the second rule. In case the second rule also cannot find a place for the next room, it is forced to insert the premature dungeon end in the place of the current room. The main goal of this work was to extend the base method by additional rules which could possibly improve the results measured in terms of generation more "interesting" and still possible to complete dungeons in shorter times.

The Return Method is proposed in place of the mechanism of average centre point used in the second rule in the base method. Instead of calculating the center

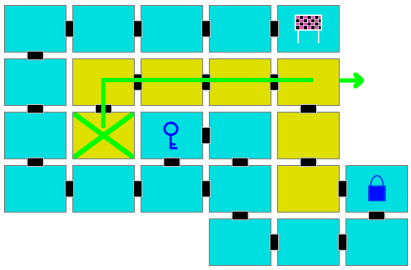


Figure 1. Return Method. The path back to finding a room that is adjacent to an empty space is marked in green. The considered subsection has yellow colour-coded rooms.

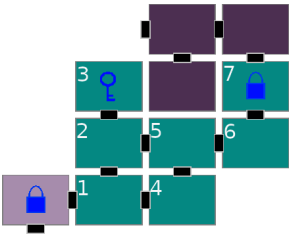


Figure 2. Restart Method. The numbers represent the rooms sequentially generated by the algorithm. The Restart Method caused the room generated as the fourth one (labeled '4') to be adjacent to the first room in the subsection ('1').

point and searching for the nearest empty space adjacent to an existing room from the current subsection, a history of movements has been introduced. Thanks to it, the algorithm can go back to the previous room to check if there is empty space around it (Fig. 1). This process of backtracking through the history of moves is repeated until a room that meets the criteria is encountered. If the generator has the possibility to choose from more paths than one, the choice is made randomly with equal probability in each of the possible directions. In the case of the proposed method, as in the base method, there is still a possibility that, despite the use of the Return Method, it will not be possible to find an empty space in the adjacence of any room included in the subsection. Then the forced end condition is also met.

The Restart Method. Placing rooms in sequence in a dungeon can lead to a high linearity of the generated dungeon. In order to reduce this linearity, a modification was proposed. Its task is to restart the generator movement from the beginning of the currently created subsection and continue generating the dungeon from this position, in case there is a key in the newly placed (current) room.

3. Experiments and Results

Criteria that are helpful to compare proposed method with the base one [2] are universal aspects such as: (1) Possibility for the player to complete the entire dungeon. (2) Measurement how "interesting" the dungeon is. (3) Measurement of

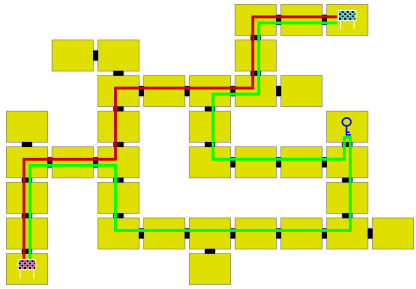


Figure 3. The difference between paths used in Path Difference heuristic: Critical – green and Spine – red.

generation time. The second aspect needs an experiment to see if the dungeon is "interesting". One of the factors that influences players' perception of a level is its linearity. The interesting path is one where the player has the choice. The Path Difference heuristic was used to assess dungeons [2]. It is based on comparing the lengths of two paths: *Critical* path and *Spine* path (Fig. 3), and is calculated according to the formula (1):

$$h = Length_{Critical} - Length_{Spine} \tag{1}$$

The *lenghts* are the numbers of rooms included in each path. The Critical path indicates which rooms the player must reach in order to complete the dungeon, remembering that a key is required to open each locked door. This is any shortest path that meets this criterion [2]. The Spine path (the shortest possible path) leads from the beginning to the end of the dungeon, without taking into account the keys.

Table 1. Completness results for 1000 generated dungeons.

Method	Complete dungeons	Forced end
Base	100%	4.9%
Average centre point and Restart Method	100%	4.8%
Return Method and Restart Method	100%	4.9%

The results in Table 1 indicate both our modifications meet the condition of completeness. Moreover, it can be concluded that proposed methods also do not increase the percentage of forced ends in the total number of dungeons generated.

The graph in Fig. 4 presents the ratios of the Path Difference heuristic score of the 500 generated 59-room dungeons to the frequency of their occurrence. There are two histograms, one for the base method and another for our method. The latter achieves much better PD heuristic scores. Ratings ranging from 30 to 36 are most commonly received, with the base method scoring 2 to 6. Thus the proposed method generates more "interesting" dungeons according to this criterion.

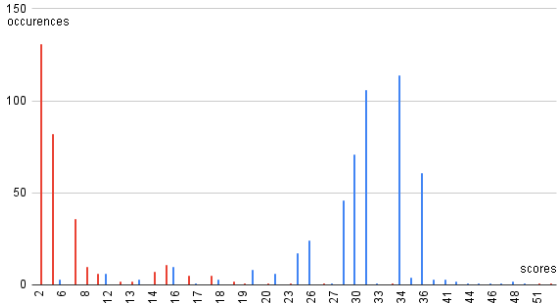


Figure 4. The ratio of the Path Difference heuristic scores of the generated dungeons to the frequency of their occurrence. Scores for paths generated with the base method are marked red and the ones for Restart Method are blue.

Table 2. Performance comparison for 59 rooms dungeons.

	Generation time in milliseconds		
Method	100 dungeons	1000 dungeons	10000 dungeons
Base	6.27	62.1	624
Proposed	4.68	47.6	475

Table 2 shows that the proposed method achieves lower average generation times compared to the base one. Figures 5 and 6 show a pair of example dungeons generated with the same seed for the pseudorandom function. The only difference between them is the method – base (Fig. 5) and proposed one (Fig. 6).

4. Conclusions

The proposed extensions: Return Method and Restart Method allowed to develop the base method to obtain better parameters of the generated dungeons and

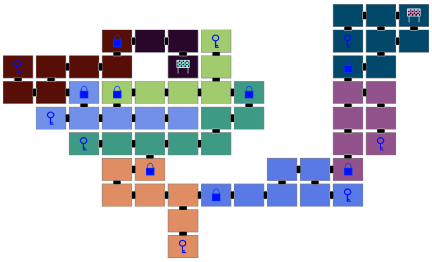


Figure 5. An example dungeon generated using the base method.

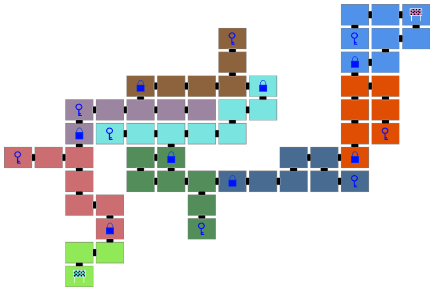


Figure 6. An example dungeon generated using the proposed method (the same seed as in Fig. 5).

the generation process itself. They yield 100% correctly generated dungeons, and in addition the Path Difference heuristic score of the dungeons created with them is much better than the score of the base algorithm. The proposed algorithm is similar to the base one in terms of the appearance of the generated dungeons and produces results that are not inferior in any way.

References

[1] Ashlock, D. Evolvable fashion-based cellular automata for generating cavern systems. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2015. doi:10.1109/cig.2015.7317958.

[2] Gellel, A. and Sweetser, P. A hybrid approach to procedural generation of roguelike video game levels. In *International Conference on the Foundations of Digital Games*, FDG '20. ACM, 2020. doi:10.1145/3402942.3402945.

[3] Thompson, T. and Lavender, B. A generative grammar approach for action-adventuremap generation in the legend of zelda. 2017. URL `https://api.semanticscholar.org/CorpusID:64591618`.

[4] Dormans, J. Adventures in level design: generating missions and spaces for action adventure games. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, FDG '10. ACM, 2010. doi:10.1145/1814256.1814257.

Finding optimal area coverage by irregular geometric items

Lyudmyla Kirichenko[1][0000−0002−2780−7993],
Sergiy Yakovlev[2,3][0000−0003−1707−843X],
Dmytro Podzeha[3][0009−0003−6629−0653], **Vadym Halii**[3][0009−0009−4059−7415]

[1]*Lodz University of Technology*
Institute of Mathematics
al. Politechniki 8, 90-924 Lodz, Poland
lyudmyla.kirichenko@p.lodz.pl
[2]*Lodz University of Technology*
Institute of Information Technology
al. Politechniki 8, 90-924 Lodz, Poland
sergiy.yakovlev@p.lodz.pl
[3]*National Aerospace University "Kharkiv Aviation Institute"*
Department of Mathematical Modeling and Artificial Intelligence
Vadima Manka 17, 61070 Kharkiv, Ukraine
{svsyak7, dima.podzega, vadim.haliy}@gmail.com

Abstract. *The paper is devoted to the study of irregular geometric covering problem. Two formulations of optimization problems are considered. The first task is to maximize the coverage of a bounded area of given sizes with a fixed number of geometric items. The second is to determine the complete coverage of the region of maximum area of a given shape. It is assumed that shapes and sizes of covering items are given. Variables of optimization tasks are placement parameters of covering items, which specify their location on the plane, as well as metric parameters of the covered area for the second task. Mathematical models of these tasks as nonlinear optimization problems are proposed. Considering the complexity of formalizing the objective function, it is recommended to use computational geometry packages, in particular Python Shapely library, to calculate its values. To solve problems, an approach based on the combined use of local and global optimization methods is described. At the local optimization stage, a so-called elastic model*

DOI:10.17388/WUT.2024.0002.MiNI

is proposed, which can significantly reduce the executed time to solve the problem. Global optimization is based on metaheuristic algorithms. The studied problems of geometric coverage have wide practical applications, in particular in territorial monitoring systems.
Keywords: *covering problem, geometric item, irregular shape, mathematical model, optimization.*

1. Introduction

Geometric coverage problem is of great interest to scientists, which is associated with wide possibilities of their practical application. First, this is due to the monitoring of bounder territories and spaces. In its informal formulation, the problem of geometric coverage is to find such a location of geometric items of various physical natures in order to maximize control over a given area. From the point of view of optimization theory, two main formulations of geometric coverage problem can be distinguished:

- to maximize the covered part of the area with a fixed number of covering items;
- to minimize the number of covering items that provide complete coverage of the area.

In the presented paper, we will consider optimization problems of geometric covering in the case when the number of covering items is fixed, and their shape and size are specified. As for the area to be covered, we will distinguish two statements depending on whether its sizes are variable.

The main difficulties in formalizing geometric covering problems are associated with the complexity of the analytical description of the dependence of the covered area on the parameters of geometric items, i.e. on their shape, location and size. An analysis of modern publications shows that their authors consider geometric items of simple shape - circles, rectangles, regular polygons. Moreover, as a rule, covering items are assumed to be equal. However, even in these cases, covering problems are NP-hard.

This paper is perhaps the first to propose an approach to formulating and solving irregular covering optimization problem of arbitrary shape geometric items.

2. Materials and Methods

2.1. Problem Statement

In d-dimensional Euclidean space R^d there is a geometric item S_0 to be covered and a family of geometric items $S_i, i \in J_n$ called covering ones. Here we use a notation $J_n = \{1, \ldots, n\}$. The problem is to determine the optimal location of items $S_i, i \in J_n$ to cover the area S_0.The specifics of such a problem depend on a objective function, geometric properties of items, as well as restrictions on their location. In this article we will consider planar geometric items S_0 and $S_i, i \in J_n$ in R^2.

Let Oxy be the Cartesian coordinate system in R^2. The shape of item S_i will be specified by the equation of its boundary $f_i(P, m^i) = 0$, $P \in R^2$ which contains metric parameters m^i characterizing the linear sizes of the item. We fix the position of item S_0 in Oxy, and for each covering item $S_i, i \in J_n$ give an internal point c_i, which will called a pole. The location of S_i will be determined by coordinates (x_i, y_i) of pole c_i and an angle θ_i of S_i rotation relative to S_0. We will call $p^i = (x_i, y_i, \theta_i)$ the placement parameters of S_i. An item S_i with placement parameters p^i we will denote $S_i(p^i)$ and call it parameterized. We put $p^0 = (0, 0, 0)$.

In this article, we will consider the following class of covering problems. We will assume that the number n of covering items $S_i, i \in J_n$ is given, they have non-empty interiors, their boundary equations $f_i(P, m^i) = 0$ are known and the metric parameters m^i are fixed, i.e. $m^i = \hat{m}^i$. An item S_0 with placement parameters p^0 and metric parameters m^0 denote by $S_0(m^0)$.

As the objective function, we choose to maximize the area of $S_0(m^0)$ covered by $S_i(p^i), i \in J_n$.

This raises two particular problems:

- *Problem 1* assumes that the metric parameters m^0 of the item $S_0(m^0)$ are fixed, i.e. $m^0 = \hat{m}^0$;
- *Problem 2*, when metric parameters m^0 are variables.

2.2. Mathematical Modeling

Let us form a parameterized complex item

$$S(\mathbf{p}, m^0) = S_0(m^0) \bigcap \bigcup_{i=1}^{n} S_i(p^i), \tag{1}$$

where $\mathbf{p} = (p^1, \ldots, p^n)$.

We introduce the characteristic function

$$\lambda_i(P, p^i) = \begin{cases} 1, & \text{if } P \in S_i(p^i); \\ 0, & \text{otherwise.} \end{cases}$$

Then function

$$\mu(\mathbf{p}, m^0) = \iint\limits_{S_0(m^0)} \Big(1 - \prod_{i=1}^{n} (1 - \lambda_i(P, p^i))\Big) dxdy \tag{2}$$

defines the dependence of the area of parameterized complex item $S(\mathbf{p}, m^0)$ on m^0 and placement parameters $\mathbf{p} = (p^1, \ldots, p^n)$.

Thus, *Problem 1* can be formulated as an unconstrained nonlinear optimization problem

$$\mu(\mathbf{p}, \hat{m}^0) \to max. \tag{3}$$

Let us denote by $\mu_0(m^0)$ the area of $S_0(m^0)$ as a function depending on metric parameters m^0. Then *Problem 2* can be formulated as follows

$$\mu_0(m^0) \to max \tag{4}$$

subject to

$$\mu(\mathbf{p}, m^0) = \mu_0(m^0). \tag{5}$$

2.3. Solving the Problems

It is very difficult to specify an explicit analytical dependence of function $\mu(\mathbf{p}, m^0)$ on placement parameters $\mathbf{p} = (p^1, \ldots, p^n)$, even if items S_0 and $S_i(p^i), i \in J_n$ have a simple shape. We propose to use *Python Shapely library* for calculations $\mu(\hat{\mathbf{p}}, \hat{m}^0)$ at the fixed $\hat{\mathbf{p}} = (\hat{p}^1, \ldots, \hat{p}^n)$. The *Shapely package* allows us to perform operations on geometric items using logical operators. As a result, from a set of basic shapes, such as a polygon, circle, ellipse, complex geometric items can be built on a plane. Since *Shapely package* runs with items of the *Polygon* class, for complex-shape items, at the preliminary stage they are approximated by polygons with a given accuracy. Python codes for calculating the area of complex-shaped objects formed by union and intersection of irregular polygons, circles and ellipses are offered in [1, 2].

Solving *Problems 1, 2* is difficult due to their large dimension and multiextremal nature. This explains the need to develop special optimization approaches that consider the features of objective functions (3), (4), and constraint (5). This approach is based on the combined use of local and global optimization methods. At the stage of local optimization, for the unconstrained optimization problem (3), we used the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method with first-order differences.

To solve *Problem 2*, we jointly used the methods of BFGS and penalty functions. At the global optimization stage, it is necessary to sort and regulate local extrema. To do this, one can use metaheuristic algorithms or guided random search. In this article, we implemented a multistart algorithm.

A computational time of methods depends on how close the starting point is to it. It is of interest to estimate the area of an complex item $S(\hat{\mathbf{p}}, \hat{m}^0)$ without implementing the union operation over all items $S_i(\hat{p}^i), i \in J_n$ formed $S(\hat{\mathbf{p}}, \hat{m}^0)$.To solve approximation problem, we propose the approach based on *Inclusion-Exclusion Principle*, limiting the number of terms of series, i.e. multiplicity of overlapping of items $S_i(\hat{p}^i), i \in J_n$.

This approach is justified when solving coverage problem with arbitrary shaped items. Indeed, it is natural to expect that when maximizing function $\mu(\mathbf{p}, m^0)$, the multiplicity of item overlapping will decrease. As a result, we formed the so-called *Elastic model* of problem (3), presenting it in the form

$$\nu(\mathbf{p}, m^0) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mu_{ij}(p^i, p^j) - \sum_{i=1}^{n} \mu_{i0}(p^i) \to min, \tag{6}$$

where

$$\mu_{i0}(p^i, m^0) = \iint\limits_{S_0(m^0)} \lambda_i(P, p^i) dxdy$$

$$\mu_{ij}(p^i, p^j) = \iint \lambda_i(P, p^i)\lambda_j(P, p^j) dxdy.$$

A special feature of model (6) is that for items shaped like a circle, ellipse, or polygon, it is possible to offer formulas for calculating functions $\mu_{i0}(p^i, m^0)$, $\mu_{ij}(p^i, p^j)$ and their gradients.

Thus, the general approach to solving *Problems 1, 2* is as follows. We randomly generate the initial locations of items $S_i(p^i), i \in J_n$ in the area $S_0(m^0)$ and estimate the values of the objective function (2), solving the local optimization

problem of function $\nu(\mathbf{p}, m^0)$. At the same time, we use an Elastic (approximate) model of the problem (6), which significantly reduces the execution time of the task. Using the multistart method, we select the best of the obtained local solutions. Then we try to improve this solution using an exact model, that is, we replace function $\nu(\mathbf{p}, m^0)$ with $\mu(\mathbf{p}, m^0)$. As a result, the optimization problem of function $\mu(\mathbf{p}, m^0)$ is solved only once, and the local solutions are sorted for simpler problems (6).

2.4. Examples

The numerical experiments have been carried out on a PC with the following configuration: CPU Intel Core i7-5557U (3.1 GHz, two cores, four threads) RAM 16 GB DDR3 1866 MHz, SSD 512 GB, OS Mac OS X11.0 Big Sur.

Let us present the results of solving *Problems 1, 2* for various shapes of the covered item and covering items. Figure 1 shows the optimal location of $n = 36$ polygons $S_i, i \in J_n$ in the rectangle $S_0(\hat{m}^0)$ of fixed sizes $\hat{m}^0$ (*Problem 1*), and Figure 2 shows the location corresponding to complete coverage of a rectangle $S_0(m^0)$ of maximum area (*Problem 2*). At the local optimization stage, the BFGS method with first-order differences was implemented for the elastic model (6). Function values $\mu_{i0}(p^i, m^0)$, $\mu_{ij}(p^i, p^j)$ and $\mu(\mathbf{p}, m^0)$ were calculated using the *Shapely package*. For global optimization, 100 starting points were generated in the multistart algorithm. The execution time for solving problems is 870 s and 1150 s, respectively.

Similarly, Figure 3 shows the optimal location of $n = 36$ ellipses $S_i, i \in J_n$ in a circle $S_0(\hat{r}^0)$ of a fixed radius $\hat{r}^0$ (*Problem 1*), and Figure 4 shows the location corresponding to complete coverage of a circle $S_0(r^0)$ of maximum radius r^0 (*Problem 2*). At the stage of local optimization for the elastic model (6), the BFGS method was implemented, in which functions and their gradients were specified using well-known formulas. For global optimization in the multistart algorithm, 100 starting points were generated. The execution time for solving the problems was 430 s and 570 s, respectively.

3. Conclusions

The paper considers two formulations of optimization problems of irregular planar coverage, in which both the coverage region and the covering items have a complex shape. The first task is to maximize the area of a bounded region of given

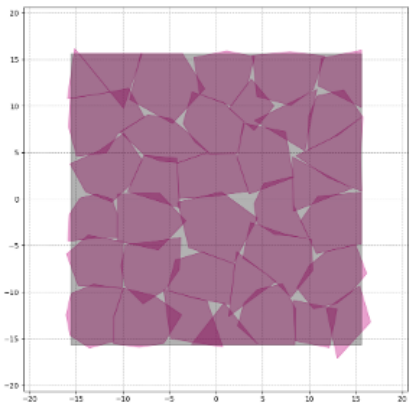


Figure 1: Maximum coverage of a square of fixed sizes by polygons

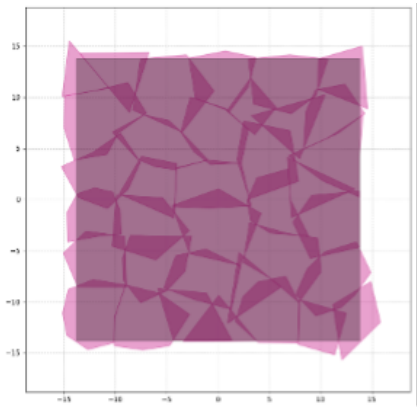


Figure 2: Complete coverage of the maximum area square by polygons

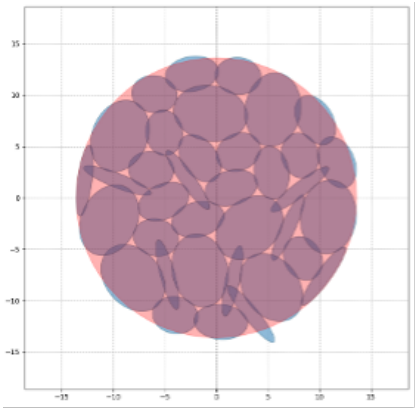


Figure 3: Maximum coverage of a circle of fixed radius by ellipses

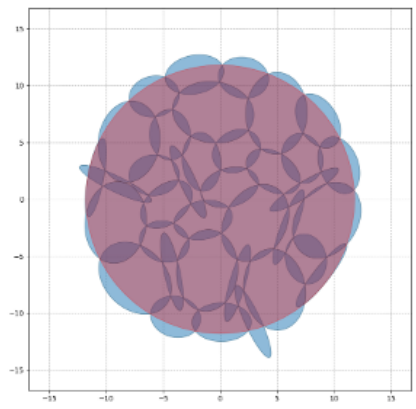


Figure 4: Complete coverage of the maximum area circle by ellipses

sizes when a number of covering items is fixed. The second task is to completely cover the region of maximum area. It is assumed that the shapes and sizes of the covering items are given. The variables of the optimization problems are the placement parameters of the covering items, which specify their location on the plane,

as well as the metric parameters of the covered region (for the second task). Mathematical models of these problems as nonlinear optimization ones are built. We propose to use computational geometry packages, in particular the Python Shapely library, to calculate values of objective function. To solve presented tasks, an approach based on the combined use of local and global optimization methods is described. At the local optimization stage, an elastic model of the coverage problem was built, which made it possible to significantly reduce the time required to solve the problem. For global optimization, the multistart algorithm was used. The authors intend to carry out further to developing metaheuristic global optimization algorithms and directed random search methods.

The studied geometric coverage problems have wide practical application, in particular, in territorial monitoring systems. In this regard, the authors plan to use the results obtained in the design and optimization of wireless sensor networks, as well as in a business site location.

References

[1] Yakovlev, S, Kartashov, O, Mumrienko, A. Formalization and solution of the maximum area coverage problem using library Shapely for territory monitoring. *Radioelectronic and Computer Systems*, 2, pages 35–48, 2022. doi: 10.32620/reks.2022.2.03.

[2] Yakovlev, S, Kartashov, O, Podzeha, D Mathematical models and nonlinear optimization in continuous maximum coverage location problem. *Computation*, 10(7):119, 2022. doi:10.3390/computation10070119.

Mitigating Dimensionality in 2D Rectangle Packing Problem under Reinforcement Learning Schema

Waldemar Kołodziejczyk[1][0000−0002−2031−4264],
Mariusz Kaleta[2][0000−0002−2225−8956]

[1] *kolodziejczykwaldemar222@gmail.com*
[2]*Warsaw University of Technology*
Faculty of Electronics and Information Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
mariusz.kaleta@pw.edu.pl

Abstract. *This paper explores the application of Reinforcement Learning (RL) to the two-dimensional rectangular strip packing problem. We propose a reduced representation of the state and action spaces that allow us for high granularity. Leveraging UNet architecture and Proximal Policy Optimization (PPO), we achieved a model that is comparable to the MaxRects heuristic. However, our approach has great potential to be generalized to non-rectangular packing problems and complex constraints.*
Keywords: *rectangular strip packing, reinforcement learning, action space size reduction*

1. Introduction

We consider a classic NP-hard problem, yet very practical in many fields, the rectangle two-dimensional strip packing problem. The set of N rectangles is to be packed as densely as possible in a strip of a given width. For simplicity, without loss of generality, we limit the height of the strip to a sufficiently large value and consider a bin of fixed dimensions throughout the paper. We focus on the online version of the problem, where rectangles are processed in descending order of their area. The typical known approaches to solving the problem include constructive heuristics and metaheuristics, among others [1, 2]. We aim to apply

DOI:10.17388/WUT.2024.0002.MiNI

the Reinforcement Learning (RL) approach, assuming a grid representation of the bin and modeling the problem as a Markov Decision Process (MDP). The grid representation means that the bin is discretized into cells, w columns and h rows, that resemble pixels, and the whole method can be perceived as vision-based. Our primary motivation is that such an approach, in contrast to known heuristics, has a great potential to be generalized to other shapes and to accommodate various constraints. Since the straightforward approach suffers from the dimensionality of the action space, we propose a representation that highly reduces the size of the space.

2. Literature Review

A literature review reveals that the practical application of machine learning, including reinforcement learning, in the domain of Bin Packing and Strip Packing is still in its early stages. Traditional approaches to solving these problems involve heuristics such as genetic algorithms [1] and Guided Local Search algorithms [2]. Recent research is primarily focused on RL applied to 3D Bin Packing Problems; however, results are obtained for relatively low granularity of the problem [3, 4, 5]. 2D Rectangular Strip Packing Problem significantly differs from 3D Bin Packing in operations, model construction, and application contexts. Directly applying Reinforcement Learning to solve 2D rectangular packing has seen limited research [6]. Nevertheless, recent attempts leveraging machine learning for 2D rectangular packing have yielded notable progress. One strategy involves employing a Convolutional Neural Network as a Q-value estimator within the framework of Double Deep Q Learning [7]. However, the effectiveness of the proposed methodology is limited by the very low state resolution of the 6x6 grid. Xu et al. use RL and pointer network for 2D rectangular strip packing problem to determine a sequence of items that are packed with MaxRects heuristic [8]. A promising direction is the integration of reinforcement learning with mathematical optimization models for packing and the exploration of hybrid RL algorithms [4, 6]. Frequently used algorithms are often not scale invariant and encounter difficulties in transferring to other similar tasks. To the best of our knowledge, this paper is the first to try to solve a high-granularity problem and tackle the problem of action space size.

3. Model of Markov Decision Process

Let the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ represent the Markov Decision Process, where $\mathcal{S}$ is the set of states that an agent can observe, $\mathcal{A}$ is the set of actions of an agent, $\mathcal{R}$ is the reward function, and $\mathcal{P}$ is probability of transition from given state to another state under given action. A pixel-like representation of the bin means large state and action spaces (any location in the 2D plane) [4]. Our experiments with such representation revealed unsatisfactory performance and convergence issues. To address this challenge, we narrowed down the degrees of freedom along the height dimension. This reduction compelled the agent to operate in a 1D environment, facilitating item placement solely based on the X-coordinate, akin to the mechanics observed in Tetris.

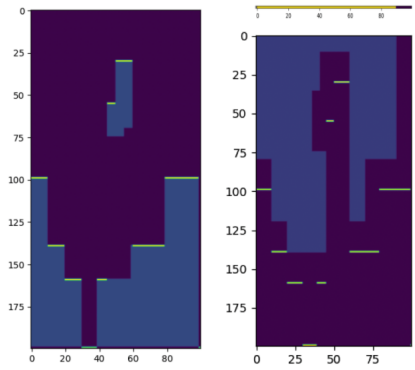
Inspired by work [4] on the 3D Bin Packing Problem, we define the state space $\mathcal{S}$ with five vectors (channels). The reduced state representation encapsulates the spatial configuration of elements in the bin, providing information about the current arrangement, possible placement locations, and details about the shape and dimensions of each element. Channel 1 is a normalized height map M representing the occupation level of the bin. For each pixel, it is the distance from the bottom of the bin to the last encountered item placed in the bin marked as a yellow bar in Fig. 1a. Channels 2-3 are binary masks indicating potential locations, so-called feasibility maps, for placing an element at two different rotation angles: 0 and 90 degrees (see a bar on the top of Fig. 1b as exemplary feasibility map for a not rotated object). Channels 4-5 are 2-element embedding representing the shape of the element, including the normalized height and width of the current element. The size of the space is $5 \cdot w$.

The action space $\mathcal{A}$ for the RL agent is a tensor composed of two vectors, channel 1 and channel 2. Each channel is a vector of policy network probabilities representing the desirability of placing an element in a specific location; however, channel 1 considers a non-rotated item, and channel 2 considers an element rotated by 90 degrees. The size of the action space is $2 \cdot w$.

In order to guide the model towards a desired outcome, we designed two versions of the reward function, (a) $V1$ containing only terminal reward, and (b) $V2$ enhanced with intermediate reward (illustrated in Figure 2):

$$\text{(a) } R_T = \frac{\sum_{n=0}^{N-1} P_n}{P_c}, \quad \text{(b) } R_t = \begin{cases} -P_L, & \text{if } t < T \\ \frac{\sum_{n=0}^{N-1} P_n}{P_c}, & \text{otherwise} \end{cases} \tag{1}$$

where t is iteration, P_n is an area of the n-th element, P_c is an area of the region (0,



(a) (b)

Figure 1: Renders of the exemplary episode after insertion of some elements. (a) Occupied 2D area marked as blue region, channel 1: occupancy vector values represented by heights of yellow bars (b) Available 2D area accessible for new item, channel 2: feasibility binary map drawn above the bin.

w, 0, y_{max}) in which elements are present, P_L lost area during step t.

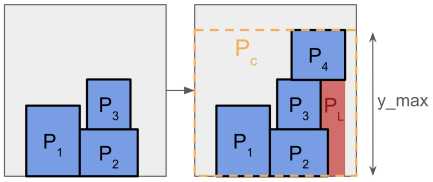


Figure 2: Reward function for 1D environment

4. Experiment Results

We tested 500 episodes, each consisting of 15 items, either fixed-sizes or randomly generated, and arranged in descending order by area, following a common practice observed in other BPP heuristics. We assumed $w = 125$ and $h = 150$. Utilizing Proximal Policy Optimization (PPO), we employed a 1D UNet architecture as the policy network for the RL agent, chosen due to its superior performance and

faster convergence compared to the equivalent Deep Q Learning alternative. The selection of the UNet architecture was motivated by the observation that determining optimal probabilities for the best action at any given time can be analogized to a classical Computer Vision segmentation task, with a focus on spatial bias and correlation among neighboring pixels. The stopping criteria for an episode included reaching the end of the item collection or encountering insufficient space for any rotation, applying to both feasibility masks.

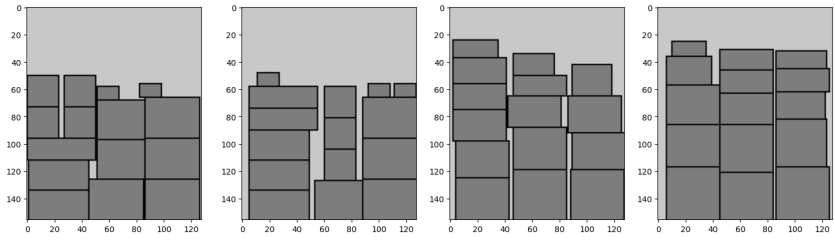


Figure 3: Exemplary results using (packed elements are white): (a) Finite set of elements with only terminal reward; (b) Finite set of elements with intermediate and terminal reward; (c) Random set of elements with only terminal reward; (d) Random set of elements with intermediate and terminal reward.

In the scenario with fixed-size elements, the agent, guided solely by the terminal reward $V1$ (see Fig. 3a), showcased an ability to plan arrangements along both the right and left borders, minimizing unused space. Figure 3b highlights a deliberate decision by the agent to leave an unoccupied spot in the middle, strategically mitigating penalties associated with the lost area. However, this strategic choice might lead to a reduced bin-filling ratio by the end of the analyzed episode. The statistical analysis of all examined episodes, when compared to the MaxRects algorithm (see Fig. 4a), indicates that the terminal reward $V1$ outperformed the intermediate alternative $V2$. The results achieved were slightly inferior to the MaxRects approach.

For the random set of elements scenario, intriguingly, the agent operating with intermediate reward $V2$ achieved superior average results when compared to the alternative utilizing only terminal reward $V1$, outperforming the MaxRects algorithm (Figure 4b). When evaluating distribution properties, it is worth mentioning that both versions, $V1$ and $V2$, exhibit smaller variances compared to the heuristic

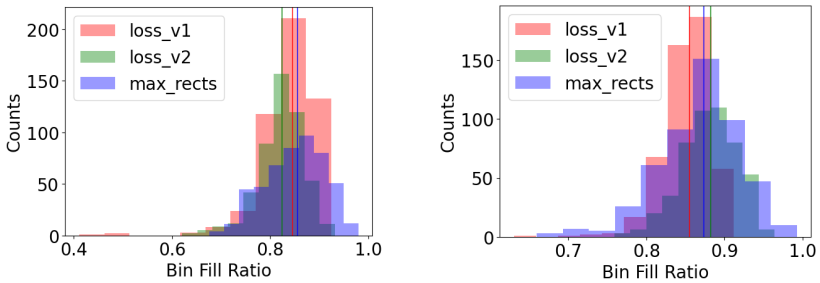


Figure 4: Histograms comparing two reward functions against the MaxRects heuristic for (a) finite set of elements, (b) random set of elements.

competitor. This indicates greater stability and reduced uncertainty in the obtained results. Analyzing the results of the exemplary episodes depicted in Figure 3c and Figure 3d, we observe that the agent tends to leave more blank space on the borders. This behavior may be a reflection of uncertainty regarding the next element, a factor that cannot be inferred from experience, as in the fixed-size variant.

5. Conclusions

While we find it hard to achieve reasonable models for full 2D bin representation, our reduced space size 1D approach under UNet-based PPO agents resulted in the model comparable to the MaxRects and even outperforming it in some cases. This achievement is promising for further investigation of non-rectangular packing. Further exploration of scenarios and variations in problem settings may provide additional insights into the capabilities and limitations of RL in the context of 2D rectangular strip packing problems. We also believe that a promising avenue lies in combining heuristics with RL-based sorting or integrating multiple heuristics.

References

[1] Hu, H., Duan, L., Zhang, X., Xu, Y., and Wei, J. A multi-task selected learning approach for solving new type 3D bin packing problem. In *Proc. of the 18th*

Int. Conf. on Autonomous Agents and MultiAgent Systems (AMMAS), pages 1386–1394. 2019.

[2] Faroe, O., Pisinger, D., and Zachariasen, M. Guided local search for the three-dimensional bin packing problem. *INFORMS Journal on Computing*, 15, 2000. doi:10.1287/ijoc.15.3.267.16080.

[3] Zhao, H., Zhu, C., Xu, X., Huang, H., and Xu, K. Learning practically feasible policies for online 3D bin packing. *Sci. China Inf. Sci.*, 65(112105), 2022.

[4] Zhao, H., She, Q., Zhu, C., Yang, Y., and Xu, K. Online 3d bin packing with constrained deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*. 2020.

[5] Puche, A. V. and Lee, S. Online 3D bin packing reinforcement learning solution with buffer. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8902–8909. 2022.

[6] Fang, J., Rao, Y., and Shi, M. A deep reinforcement learning algorithm for the rectangular strip packing problem. *PLoS One*, 18(3):e0282598, 2023.

[7] Kundu, O., Dutta, S., and Kumar, S. Deep-pack: A vision-based 2d online bin packing algorithm with deep reinforcement learning. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–7. 2019. doi:10.1109/RO-MAN46459.2019.8956393.

[8] Xu, X., Chen, J., Rao, Y., Meng, R., Yuan, B., and Luo, Q. Transfer ants reinforcement learning algorithm and its application on rectangular packing problem. *Computer Integrated Mfg. Sys.*, 26(12):3236–3247, 2020.

Anticipatory model of intelligent decision support in industrial risk management systems

Andrzej M.J. Skulimowski[1,2][0000−0003−0646−2858], **Paweł Łydek**[1,2][0000−0001−9077−4998]

[1]*AGH University, Decision Science Laboratory, 30-059 Kraków, Poland*
[2]*International Centre for Decision Sciences and Forecasting, Progress & Business Foundation, ul. Juliusza Lea 12B, 30-048 Kraków*
ams (at) agh.edu.pl, lydek (at) agh.edu.pl

Abstract. *This article presents the idea of using anticipatory network (AN) as a rescue action modelling technique in intelligent decision support systems (IDSS) for management of industrial risks and safety. IDSS link different departments of an enterprise in case of emergency and supports multiple threat management activities. This class of intelligent systems can also cover long-term resilience planning, insurance expenditure, and related financial risk management. The main idea of the presented approach consists in solving a multi-level multicriteria optimization problem embedded in ANs: adjusting the IDSS features to cyber-human capa-bilities at the higher level and ensuring optimal emergency action planning at lower levels. This model includes ad hoc organizational structures created to handle crisis situations, natural disasters, technical accidents, as well as anthropogenic threats. The above ideas have been applied to an IDSS design for a large industrial plant in the energy sector.*
Keywords: *AI alignment, Intelligent decision support systems, Anticipatory networks, AI-based action planning, Industrial risk management*

1. Introduction

Enterprise risk management (ERM) systems usually focus on financial risks, while industrial risk management (IRM) is a broader topic, covering the safety issues in cyber-physical systems. The data gathered with video surveillance and other sensors [3] is used by IRMs to define physical control and preventive measures. Both ERM and IRM systems also differ in the time frame of deployment:

DOI:10.17388/WUT.2024.0002.MiNI

ERM is used for long-term financial planning without any pre-determined particular activity of deployment. On the contrary, the relevance of IRM is particularly evident in crisis situations when the IRM system governs short-term emergency response and middle-term recovery activities [9]. In this article, we argue that due to the availability of new artificial intelligence (AI) methods, it is possible to integrate risk management in a multi-level intelligent decision support system (intelligent DSS, or IDSS) that covers strategic and operational planning, as well as provides ad hoc clues concerning the emergency response. To build and update the relevant threat models autonomously, intelligent decision support procedures are often enhanced by machine learning (ML) techniques. Threat model parameters are learned from retrieved data streams and prior emergency action records, and applied to derive optimal prevention, mitigation, and rescue activities. An intelligent DSS endowed with ML tools is capable of recommending situation-dependent risk mitigation actions, operations and strategies to ensure an optimal level of industrial safety for all planning horizons applicable in an enterprise. Such systems are termed industrial risk management DSSs (IRM DSS [10]).

The ideas presented in this article have been inspired by the real-life case of an IRM DSS designed for a large industrial plant, where heterogeneous natural and anthropogenic threats can be mitigated with a combination of AI-based techniques, such as ML, sensor information fusion, image understanding, and multicriteria decision support procedures.

To integrate risk management at different levels efficiently within all relevant planning horizons, from immediate remedies to planning complex operations and long-term strategic resilience-building measures, we propose a causal model of threats, risks, crisis management, and their consequences. Supplemented by anticipatory decision making [7], [5]. The ultimate IRM model has been applied to support emergency and resilience building decisions at all applicable levels.

The primary aim of this article is to propose a decision support approach and the related IRM DSS software architecture that allows decision makers responsible for crisis situation management to integrate the surveillance, signal processing, and decision support technologies into a holistic industrial security system based on an anticipatory network (AN) model.

2. Industrial risk management problem statement

In the context of industrial security problems presented in this article, risk is attributed to external threats, to information processing procedures that can bias threat observations with errors, to human operational errors, and to systematic erroneous decisions that can be made during risk management. The threat data transfer can be modelled as a network, where information loss and incidental operational and decision-making faults are sources of additional risks. This network is complemented by the risk management and optimization model involving decision algorithms, actions, and actors implementing them. Both components of the model are coupled by feedback information, received by the sensors, compared to the values provided by the model and presented to the ML module supervised by the emergency situation management acting as decision makers.

The industrial risk analysis quantifies the potential threat impacts, discerns the relationships between them and recommends optimal risk prevention and mitigation measures. The availability of information stored and processed in other enterprise information systems and the communication efficiency are crucial factors for the deployment of an IRM DSS [9],[10].

While AI is widely used in financial risk management systems, its deployment in natural and anthropogenic risk management systems was rare up until recently [8]. The contemporary crisis management systems with decision support functionalities emerged from early warning systems that evolved into cloud-based heterogeneous signal processing [4]. Advanced ML-based information processing software allow DSS developers to effectively integrate IRM systems within ICT infrastructure of an enterprise. Further state-of-the-art AI methods used in IRM such as automatic dangerous situation recognition in visual monitoring data [1] can satisfactorily meet the enterprise safety goals. A general model of IRM contains a network of information processing modules, which includes sensors, information fusion, decision support, as well as automated decision-making nodes, which all form a *knowledge graph* [10], a key structure in IRM DSS. The resulting decision support model based on knowledge graphs contains also the first responder teams, their action execution hierarchy, knowledge about other objects, and exchange of this information.

The latter objects, together with the safety instructions, execution orders, expectations, and causal relations between them can be represented as a dynamic directed multigraph, with three types of edges that denote information flow, threat propagation and impact, as well as commands transmission. Such multigraph, with

nodes corresponding to agents and active objects is an example of anticipatory network (AN) [7], which will be defined in the next section.

The decision problem to be solved with the above hybrid hard- and software model can be formulated as follows.

Problem 1. Assume that N multicriteria decision problems $O_1, \ldots, O_N$ are embedded in a hierarchical reporting structure that forms an acyclic connected digraph with one root. Each O_i, $i = 1, \ldots, N$, is solved by a decision maker D_i. According to general emergency management instructions D_i selects decisions from a given set U_i so that the criteria $F_i := (F_{i1}, \ldots, F_{ik(i)})$ are optimized on U_i and an additional preference information P_i concerning the selection of a nondominated decision from U_i is taken into account. In addition, by selecting a decision u_i, the decision maker D_i that precedes D_j in the reporting order can impose additional constraints $\psi_{ij}(u_i)$ on the decisions of D_j. When transferring a direct command from the decision maker D_i to D_j is impossible, D_i may want D_j to select an element of the set V_{ji}, which - according to the D_i's knowledge – contains decisions that may be beneficial to the course of emergency action. The deontic condition 'u_j ought to belong to V_{ji}' is termed anticipatory feedback f_{ji} (AF). A strategy for all decision makers D_i with a successor should be found to satisfy the maximum number of AFs in the network, or to optimize another objective that depends on all AF's. ■

3. Causal and anticipatory decision models in IRM DSS

To optimize quantitative crisis management criteria F_i in the above Problem 1 we applied anticipatory decision making algorithms embedded in an AN. According to the definition presented in [7], AN is a directed multigraph composed of acyclic causal sub-digraphs defined by the causal relations ψ_{ij} and the anticipatory feedbacks f_{ji}. Moreover, it is assumed that each pair of nodes (A, B) connected by the AF satisfies the implication

$$Af_{ji}B \Rightarrow A\Psi B, \tag{1}$$

where Ψ is the transitive closure of the relation ψ_{ij}. i.e. $A\Psi B$ iff there exists a path from A to B in the graph of ψ_{ij}. An example of an anticipatory network modelling a real-life decision structure in Problem 1 is presented below.

Example. Figure 1 below shows a realistic AN that refers to solving Problem 1 with the IRM DSS designed for a limestone mine in Poland. This instance of the

anticipatory decision making problem with 9 nodes and 7 AFs was solved with dedicated AN software available at www.anticipatorynetworks.net. The nodes $D_i, i = 0, 1, \ldots, 8$, correspond to decision makers, including the crisis manager D_0 and response teams D_6, D_7 (both internal), and D_8 (an external rescue team). The other nodes in the network model intermediate crisis management level (D_1 and D_4) and the supplementary or reserve teams (D_2, D_3, and D_5). The causal graph (red edges annotated ψ_{ij}) models the reporting relation and commands issued by decision makers. The notation $D_i\psi_{ij}D_j$, depicted in Figure 1 as an edge from D_i to D_j, is equivalent to the definition by D_i additional constraints on decisions allowed to be made later by D_j. This is denoted as

$$\psi_{ij}(u_{i,k}) = \{u_{j,p1}, \ldots, u_{j,p(k)}\} \subset U_j, \tag{2}$$

where $u_{i,k}$ is a decision selected by D_i from the set U_i, and U_j is the set of all admissible decisions of D_j. For the rescue teams, the elements of U_i correspond to real-life threat mitigating activities on endangered objects $Z_1, \ldots, Z_m$, for example 'go to Z_k', 'go halfway to Z_k', for certain $1 \leq k \leq m$, 'pursue action till the evacuation end' 'withdraw' etc. Thus, the relation ψ defines the decision problem hierarchy, where the decisions $u_{i,k}$ made at higher level impact decision makers $D_{j1}, \ldots, D_{jm}$ at lower level according to (2), provided that D_i is linked by ψ_{ijk} with D_{jk}. Unequivocal commands correspond to $p(k) = 1$, but this is an exceptional situation, as the main idea behind the anticipatory decision making structure is to provide subordinated decision makers with a certain level of freedom in unexpected circumstances. This freedom is transferred to lower-level units, rescue teams and autonomous robotic actuators.

The second subgraph (blue edges annotated f_{ji}) models AFs, specifically each f_{ji} defines the set $V_{ji} \subset U_j$ that contains decisions solicited by D_i. The sets V_{ji} contain usually actions which are not mandatory for D_j, nevertheless can be beneficial for the overall emergency activity, e.g. V_{ji} may be defined as '*keep on fire extinguishing*', '*refrain from withdrawal, even if it is allowed*'. The V_{ji} rarely coincide with any of the constraints $\psi_{ij}(u_{i,k})$, so D_i strives to choose a decision that will make D_j choice of an element of V_{ji} most probable.

Assuming that D_j is *rational*, i.e. selects nondominated decisions maximizing a utility function ξ_j, to find the best decision in U_i it suffices that the decision maker D_i is able to forecast the utility function of D_j. Namely, D_i should maximize the conditional probability that D_j selects a decision from the set V_{ji}. This is equivalent to maximizing in U_i the quotient function (3):

$$q(u_{i,k}) := \mu(V_{ji} \cap argmax\{\xi_j(v) : v \in \psi_{ij}(u_{i,k})\})/\mu(\psi_{ij}(u_{i,k})\}, \tag{3}$$

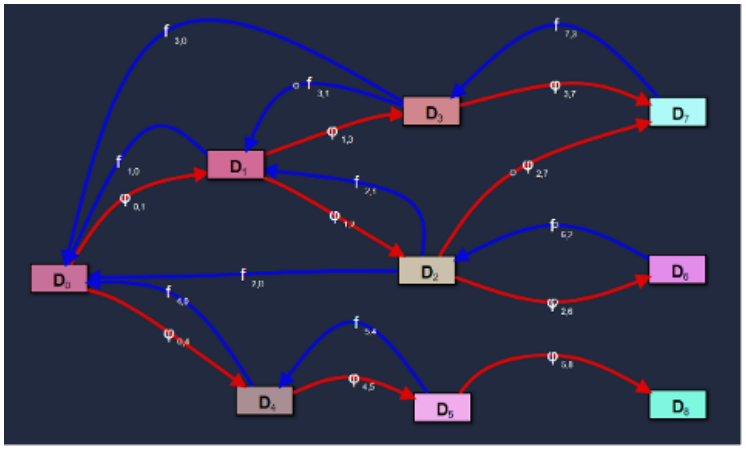


Figure 1. Anticipatory network applied to the design of decision making procedures in an IRM DSS, created with www.anticipatorynetworks.net.

where $\mu(Y)$ is a probabilistic measure of the set $Y \subset U_j$, and $argmax\{\xi(v) : v \in X\}$ denotes the subset of X where the values of the function ξ are maximal in X. According to the rationality assumption, if the set U_j is finite, one can admit that $\mu(Y)$ is the cardinality of Y divided by the cardinality of U_j.

The above approach to selecting decisions in ANs is termed *simulation principle*. Decision flow simulation and the resulting optimal decision sequences for each $D_i, 0 \leq i \leq 8$, in the AN depicted in Figure 1 are shown in the tables available for this example at www.anticipatorynetworks.net. The optimality principle applied in the decision simulation algorithm assumed that the maximum number of satisfied feedbacks should be reached. ■

Observe that Problem 1 is, in fact, a multi-level multicriteria optimization task [6], [11], where the additional constraints in a problem $(U_j, F_j)\rightarrow$min are imposed by D_i's preceding D_j in the causal order ψ. The IRM DSS design principle behind anticipatory modelling consists of designing a command structure $\psi_{i,j}$ so that AFs f_{ji} are satisfied to the maximum possible extent, for example, by a highest number of them or as a sum of positive weights for each satisfied AF and negative penalties for all non-satisfied AFs. An optimal sequence of decisions at each level could then be found with the selected solution method. In-depth quantitative analysis of decision scenarios in ANs with different coordination levels expressed by the

functions ψ_{ij} are a subject of current research.

4. Conclusions

Industrial risk management systems are no longer stand-alone applications. They are predominantly hybrid cyber-physical systems deploying a variety of AI techniques and AI-enabled hardware. In addition, human-in-the-loop is an essential paradigm, paving the ways the IRM DSS are designed and deployed. The experience gained when selecting AI methods to support rescue activity planning in a large energy-sector enterprise allowed us to detect several salient development trends. First of all, due to the growing uncertainty about the anthropogenic risks and the potential occurrence of heterogeneous multi-threats, the complexity of action planning in case of emergency grows and it becomes impossible to include all cases of potential threats and their combinations in emergency instructions. The ongoing transition from systems focused on information provision, visualization and presentation to decision makers to intelligent, partly autonomous DSSs has been also identified from bibliographic scans as global trends. These have been already taken into account at the IRM DSS design stage.

The causal and anticipatory decision models presented in this article have proven to be particularly useful decision support tools when planning res-cue activities in case of natural disasters. Adaptive decision rules embedded in ANs can compensate for communication breakdowns and make the emergency instructions flexible, yet conforming to the law and general corporate safety policies. Further relevant AI methods which have been implemented in IRM DSSs or applied in the design process include reinforcement and semi-supervised ML procedures applied to learn threat models and their resilience responses [2], decision rules, and the parameters of mitigating activities for the emergency situation [1]. The conflicts between the objectives in the emerging multicriteria optimization problems could be efficiently resolved with decision support methods based on multiple classes of reference sets [10]. When applied in the DSS interactive procedure design aimed at solving industrial risk management problems in real time, the reference sets can ensure intuitive and efficient communication schemes with human risk managers and emergency action supervisors.

References

[1] Foresti, G.L., Marcenaro, L., and Regazzoni, C.S. Automatic detection and indexing of video-event shots for surveillance applications. *IEEE Trans. Multimedia* 4(4), 459-471, 2002. doi:10.1109/TMM.2002.802024.

[2] Gong, J. and You, F. Resilient design and operations of process systems: Nonlinear adaptive robust optimization model and algorithm for resilience analysis and enhancement. *Comp. Chem. Engrg*, 116, 231- 252, 2018.

[3] Mehla, S. and Jain, S. An ontology supported hybrid approach for recommendation in emergency situations. *Ann. Telecomm.*, 75, 421–435, 2020.

[4] Middleton, S. E., Middleton, L., and Modafferi, S. Real-time crisis map-ping of natural disasters using social media. *IEEE Intell. Syst.*, 29(2), 9-17, 2014. doi:10.1109/MIS.2013.126.

[5] van Niekerk, D. and Terblanché-Greeff, A. Anticipatory Disaster Risk Reduction. In Poli, R. (ed.) *Handbook of Anticipation*, pages 1659-1681, Springer, Cham, 2019. doi:10.1007/978-3-319-91554-8_90.

[6] Pfetsch, M.E. and Schmitt, A. A generic optimization framework for resilient systems. *Optimization Methods and Software*, 38(2), 356-385, 2023.

[7] Skulimowski, A.M.J. Anticipatory Network Models of Multicriteria Decision-Making Processes. *Int. J. Systems Sci.*, 45(1), 39-59, 2014. doi: 10.1080/00207721.2012.670308.

[8] Skulimowski, A.M.J. and Bañuls, V.A. AI Alignment of Disaster Resilience Management Support Systems. In Rutkowski, L. et al. (eds), *Artificial Intelligence and Soft Computing. Lecture Notes in Artificial Intelligence* 12855, Springer, pages 354-366, 2021. doi:10.1007/978-3-030-87897-9_32.

[9] Skulimowski, A.M.J. and Łydek P. Applications of AI Alignment and Anticipatory Networks to Designing Industrial Risk Management Decision Support Systems. In Buchmann R.A. et al. (eds.), *Proceedings of the 30th International Conference on Information Systems Development (ISD2022)*, Cluj-Napoca, Romania, 31 August–2 September, 2022, pages 1-6 (2022). https://aisel.aisnet.org/isd2014/proceedings2022/ai/2/

[10] Skulimowski, A.M.J. and Łydek P. Adaptive Design of a Cyber-Physical System for Industrial Risk Management Decision Support. In *Proceedings of the 2022 IEEE 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, Dec. 11-13, 2022, IEEE CPC Press, pages 90-97, 2022. doi: 10.1109/ICARCV57592.2022.10004251.

[11] Zewde, A.B. and Kassa, S.M. Hierarchical multilevel optimization with multiple-leaders multiple-followers setting and nonseparable objectives. *RAIRO - Operations Research*, 55(5), 2915 – 2939, 2021. doi: 10.1051/ro/2021146.

Chapter 10

Artificial Intelligence in Bioinformatics

Track Chairs:

- prof. Dariusz Plewczyński - Warsaw University of Technology
- prof. Tomasz Gambin - Warsaw University of Technology
- prof. Robert Nowak - Warsaw University of Technology

Prediction of amino acids mutations in SARS-CoV-2 spike protein using neural networks

Filip Korzeniewski[1], **Tomasz Gambin**[1][0000−0002−0941−4571]

[1] *Warsaw University of Technology, Institute of Computer Science Nowowiejska 15/19, 00-665 Warsaw, Poland*

Abstract. *This study addresses the need for effective SARS-CoV-2 vaccines by focusing on predicting mutations in the spike (S) protein, crucial for vaccine development. We developed a method that predicts which mutations are likely to occur in the next periods using historical spike protein sequence data. The main challenges were to process the data SARS-CoV-2 strains in a way it can be later processed by Recurrent Neural Network and to create appropriate models based on RNN able to predict mutations. Results indicate high performance of proposed method in epitope mutation prediction and further confirm the potential of machine learning in COVID-19 vaccine development.*

Keywords: *bioinformatics, SARS-CoV-2, neural networks, attention*

1. Introduction

SARS-CoV-2 is the virus responsible for the COVID-19 pandemic that began in 2019 [1]. The virus is made up of several key components, including the spike (S), envelope (E), membrane (M), and nucleocapsid (N) proteins [2]. The spike protein, in particular, is crucial for the virus's ability to infect host cells, making it a primary focus for vaccine development efforts. This is because the spike protein facilitates the virus's entry into cells by binding to the host's receptors [3].

Research indicates that the most effective way to combat the virus is to target it early in its infection process. There are two main strategies for preventing the virus from entering cells: (i) blocking the interaction between the virus's Receptor Binding Domain (RBD) and the host cell's ACE2 receptor, and (ii) destabilizing

DOI:10.17388/WUT.2024.0002.MiNI

the prefusion conformation of spike protein upon binding antibodies with epitopes, which are the parts of the virus recognized by antibodies [2, 4].

This study specifically addresses the second strategy of targeting epitopes. Vaccination prompts the immune system to produce a targeted response against these epitopes [5]. However, if the virus's epitope mutations occur, vaccines may become less effective, highlighting the importance of focusing on these components in vaccine development [4].

2. Materials and methods

The research methodology described involves a process for predicting future mutations in the epitopes of virus sequences (Fig. 1a), drawing inspiration from a method outlined in previous influenza virus study [7]. The approach is structured around three main phases. **Feature Engineering**: This initial phase involves filtering out low-quality or ambiguous sequences, clustering sequences from similar time periods to identify patterns, and linking these clusters across consecutive time periods to track the evolution of the virus. **Model Training**: Utilizing the prepared data, the next step involves training neural network (NN) models, specifically recurrent neural networks (RNNs), to predict the likelihood of mutations in specific epitope residues based on historical mutation data and the context of surrounding residues. **Model Performance Comparison**: The final phase compares the predictive performance of different models, including variations with and without attention mechanisms, and benchmarks against simpler models like logistic regression to evaluate effectiveness. To evaluate various models we consider virus genomic sequences from various time frames (from t to $t + 9$). It processes these sequences to predict whether mutations will occur in $t + 10$ time period in specific epitopes or any epitope residue. Correct predictions are counted as true positives.

For this project, adaptations were made for the coronavirus, with a more stringent filtering process compared to the influenza study. The clustering used the K-means algorithm, supported by dimensionality reduction techniques like PCA and t-SNE for visual analysis and the elbow method for determining cluster numbers. The study then employed an dual-attention-based Tempel model (Fig. 1d) for prediction, comparing it against two other RNN models, including those with a single attention (Fig. 1c) and without an attention mechanism (Fig. 1b), and evaluating a logistic regression classifier as a baseline.

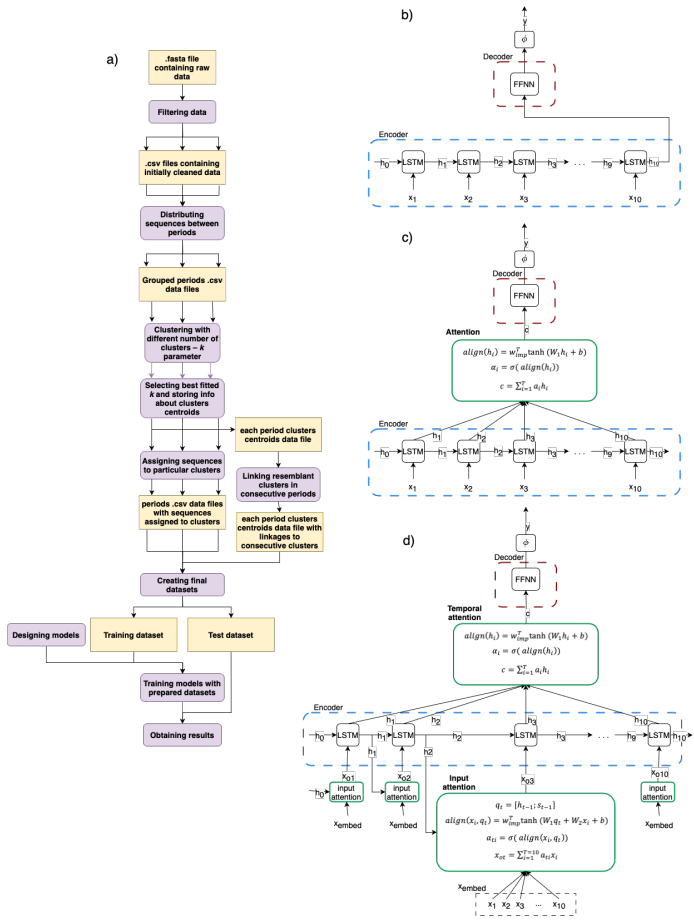


Figure 1. a) An overview of the pipeline for future mutation prediction in the epiptope of SARS-CoV-2 spike protein. All major data structures are marked with rectangular figures and yellow colors. All major activities are marked violet and rounded figures. b) Architecture of the I model: no attention mechanism. The encoder result is directly provided to decoder. c) Architecture of the II model: classic (Bahdanau et al. [6]) attention mechanism. The most relevant hidden states are selected, and their results are passed to the decoder. d) Architecture of the III model: 2 attention mechanisms. Input attention evaluates importance of features in the embedded vector sequence. Temporal attention evaluates importance of hidden states. The attention mechanisms are applied hierarchically.

2.1. Data pre-processing and model configuration

The dataset was sourced from the GISAID database [8] on March 8, 2024. This dataset, downloaded in FASTA format, focused solely on spike proteins, with an average length of 1273 amino acids, and comprised a total of 33,263,788 records at that time. The final dataset included 38,499 training, 11,759 validation, 7,841 test unique samples. Within these refined datasets, 14.79% (training), 13.22% (validation), 12.76% (test), were identified as mutated.

To transform SARS-CoV-2 cleaned protein sequences into embedded vectors, the dictionary mapping 3-grams proteins into 100-dimension vectors was used [8].

All the models consists of an encoder, decoder and 1 hidden layer. The final models' configurations is as follow: (i) RNN (no attention): hidden layer size (HLS) - 256, input layer dropout probability (ILDP) - 20%, learning rate (LR) - 0.001, mini-batch size (MBS) - 256; (ii) RNN (single attention): HLS - 512, ILDP - 20%, LR - 0.001, MBS - 256; (iii) Tempel: HLS - 256, ILDP - 20%, LR - 0.01, MBS - 256. Only CPU was used for training.

3. Results

Our analysis revealed that RNN-based models outperformed the logistic regression model in predicting mutations of the SARS-CoV-2 virus (see Table 1). Among the RNN models, the one based on the Tempel algorithm not only matched the others in accuracy but also led in recall, indicating its superior ability to identify novel and significant mutations without missing them. This model also scored the highest in terms of the F-score and Matthew's correlation coefficient (MCC), making it the most effective overall.

Table 1. Metrics achieved by all models for the SARS-CoV-2 mutation prediction task. LR – logistic regression (baseline); No-attn – no attention mechanism applied; Attn – classic Bahdenau [6] attention applied; Tempel: dual-attention mechanism (input and temporal) applied.

	accuracy	**precision**	**recall**	**mcc**
LR	0.924	0.889	0.484	0.622
No-attn	0.963	0.974	0.726	0.823
Attn	0.972	0.975	0.803	0.870
Tempel	0.977	0.977	0.843	0.895

Table 2. Mutation prediction results of logistic regression and Tempel [7] between influenza virus (H1N1) [7] and SARS-CoV-2 spike protein (this study).

model	virus type	accuracy	precision	recall	mcc
LR	**H1N1**	0.887	0.678	0.380	0.435
LR	**SARS-CoV-2**	0.924	0.889	0.484	0.622
Tempel	**H1N1**	0.955	0.853	0.821	0.806
Tempel	**SARS-CoV-2**	0.977	0.977	0.843	0.895

Our findings show that the task of predicting mutations in the SARS-CoV-2 spike protein outperforms the results of the study on influenza [7] by approximately 10-11% (see Table 2). This indicates a significant improvement in accuracy for the SARS-CoV-2 spike prediction compared to the influenza study.

4. Conclusions

This study final goal was to assess whether using deep learning, specifically recurrent neural networks, is possible to predict future mutations of SARS-CoV-2 spike protein.

Although it is not directly comparable, we found that adapted approach performs better on COVID-19 than on the influenza data set. A possible explanation for this could be that the structure of the SARS-CoV-2 spike protein is simpler for algorithms to analyze and identify patterns in, even though it is longer (with a sequence of 1273 amino acids) compared to the H1N1 spike protein, which has a shorter sequence of 566 amino acids. Additionally, when analyzing SARS-CoV-2 data using 1-month-gap windows, the differences observed between sequences over subsequent months are smaller compared to the differences observed between sequences over subsequent years when analyzing influenza datasets. Thirdly, there was a significant difference in data preparation between the two approaches. In this study, we applied a meticulous data filtration process, which likely contributed to more effective clustering and linking. By preparing the data more thoroughly, we may have enhanced the overall performance of the algorithms.

The results clearly demonstrate the feasibility of using NN models to predict SARS-CoV-2 mutations in future time periods, with performance significantly surpassing that of the baseline method.

This study highlights a promising approach for future mutation prediction efforts. There is considerable potential for enhancing the accuracy of predictions.

Only a few promising models were developed, fine-tuned, and evaluated to test our hypothesis. By refining these models—for instance, by incorporating transformer-based attention mechanisms or employing more sophisticated models better suited for this task—we could unlock even more of this approach's potential. Additionally, it would be worthwhile to explore how adjustments to the window size (number of periods), either by reducing or extending it, and altering the gap between consecutive sequences—increasing it from a monthly basis—impact the quality of the results.

References

[1] Velavan, T. et al. The covid-19 epidemic. *Tropical Medicine & International Health*, 25(3):278–280, 2020.

[2] V'kovski, P. et al. Coronavirus biology and replication: implications for sars-cov-2. *Nature Reviews Microbiology*, 19, 2020.

[3] Chen, C. et al. Computational prediction of the effect of amino acid changes on the binding affinity between sars-cov-2 spike protein and the human ace2 receptor. *bioRxiv*, 2021.

[4] Sikora, M. et al. Computational epitope map of sars-cov-2 spike protein. *PLOS Computational Biology*, 17(4):1–16, 2021.

[5] Noor, R. et al. Covid-19 vaccines: their effectiveness against the severe acute respiratory syndrome coronavirus 2 (sars-cov-2) and its emerging variants. *Bulletin of the National Research Centre*, 46, 2022.

[6] Bahdanau, D. et al. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 2014.

[7] Yin, R. et al. Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks. *Bioinformatics*, 36(9):2697–2704, 2020. ISSN 1367-4803.

[8] Khare, S. et al. Gisaid's role in pandemic response. *China CDC Weekly*, 3:1049–1051, 2021.

Chapter 11

Generative Artificial Intelligence

Track Chairs:

- prof. Maciej Zięba - Wrocław University of Science and Technology
- prof. Tomasz Trzciński - Warsaw University of Technology
- prof. Przemysław Spurek - Jagiellonian University

Image-based soundtrack generation with diffusion model

Dawid Wolkiewicz[0009−0003−6495−6167], **Piotr Syga**[0000−0002−0266−5802]

Wrocław University of Science and Technology
Faculty of Information and Communication Technology
wybrzeże Stanisława Wyspiańskiego 27, 50-370 Wrocław, Poland
259090@student.pwr.edu.pl, piotr.syga@pwr.edu.pl

Abstract. *The audio and visual stimuli are intertwined, when shown an image, one can select a complementing music sample. We propose a conditional latent diffusion model for generating new and unique audio fragments with emotional correlation to source images. The results were evaluated using objective metrics and subjective opinions in the conducted survey, achieving a mean opinion score of* 3.407(±1.155)*. This showed that the developed solution could be applied in personalized soundtrack generation, with an application in music therapy, through a contextually appropriate combination of images and soundscapes. Model structure and weights*[1]*, and an example application*[2] *were published on HuggingFace.*
Keywords: *soundtrack generation, artificial intelligence, diffusion model*

1. Introduction

Music holds profound significance in human life, influencing concentration, motivation, and emotional well-being. An essential application of music lies in its therapeutic potential, becoming integral to diverse treatment approaches. However, challenges arise in selecting music tailored to individual preferences and experiences, compounded by copyright limitations hindering the use of commercial works in therapy. We explored the intersection of computer science and music composition to address these issues, focusing on image-based soundtrack generation. In this paper, we provide an audio generation method that leverages the

[1]https://huggingface.co/Woleek/clMusDiff
[2]https://huggingface.co/spaces/Woleek/image-based-soundtrack-generation

DOI:10.17388/WUT.2024.0002.MiNI

image-conditioned diffusion model and analyzes its audio quality and emotional congruence between input image and output audio.

2. Previous works

Various approaches to sound generation from images include visual localization of sound sources and generating sounds based on identified pixels [1]. Another approach involves determining image areas explaining a given sound [2].

An emerging method combines image captioning and text-to-music synthesis, e.g., the BGT (BLIP-GPT2-TeleMelody) pipeline [3]. Alternatively, some methods directly convert image features into music sequences, like an LSTM-based model [4]. Pix2Pitch uses cGANs to transform images into Mel spectrograms [5]. A diffusion-based model from the *audio-diffusion* package synthesizes new soundtracks using audio-encoded vectors as guidance.

This paper aims to generate music that evokes specific emotions, building on previous investigations. Integrating music into therapeutic sessions, especially using Guided Imagery and Music (GIM) and Schema Therapy (ST), has been recognized for enhancing emotional expression and regulating emotions [6, 7]. Note that such research operated on various datasets, hence are not directly comparable.

3. Proposed method

Dataset Due to the intended therapeutic purposes, the selection of available datasets was focused on datasets comprising image-audio pairs with emotional matching of samples in both modalities. We examined several open-source datasets and chose the Emotionally paired Music and Image Dataset (EMID) [8]. The dataset consists of 32,214 image-audio pairs, obtained by emotionally matching each of the 10,738 unique audio fragments with three non-unique images from a separate source. Each audio sample is 5 seconds long with a sampling rate of 44.1 kHz. The images are in various resolutions, with a mean of 449x490.

The model We based our solution on an architecture[3] from the audio-diffusion package and modified it to use image conditioning that served as a connection between visual and audio stimuli. For the feature encoder component, we selected

[3]https://huggingface.co/teticio/conditional-latent-audio-diffusion-512

a Vision Transformer[4] (ViT). The model was trained on the emotionally matched image–spectrogram pairs obtained from EMID, with 90% of the samples used for training and 10% as the validation set. The training was performed for 100 epochs on a single A100 NVIDIA GPU, with mean squared error loss employed.

Our Conditional Latent Diffusion model, named clMusDiff (cf. Fig. 1), leverages the reverse diffusion process and extracted image embeddings to generate Mel spectrograms, which can then be converted into audio. The soundtrack generation is initiated by providing an image, specifying the number of denoising steps (set to 50 after empirical evaluation), and generating a random noise size-matched to the spectrogram. During the image encoding, the input undergoes encoding by the ViT module. In denoising scheduling, a vector of time steps is generated, which is crucial in controlling the denoising. Next, during latent mapping, the initial noise is transformed into a probabilistic distribution to reduce the size of the data and accelerate the generation. At every denoising step, the model predicts the noise to be removed from the sample, which is scaled by the scheduler. Finally, the scaled noise is applied to the sample, creating a less noisy version. The final latent representation is reconstructed into a Mel spectrogram using VAE. The output audio is retrieved by inversion of the spectrogram.

4. Evaluation

Objective Metrics For the objective evaluation of the generated audio, we used 3 metrics. Spectral Flatness (SF) measures the tonality of a sound (cf. Eq. 1). Zero-Crossing Rate (ZCR) depicts the rate at which the audio signal changes its sign (cf. Eq. 2). Whereas the average power of an audio signal is measured by Root Mean Square Energy (RMS), depicted in Eq. 3 .

$$\mathrm{SF} = \frac{\sqrt[N]{\prod_{n=1}^{N} x(n)}}{\frac{1}{N}\sum_{n=1}^{N} x(n)}\,, \tag{1}$$

where N is the number of frequency bins, and $x(n)$ is the magnitude of frequency bin n.

$$\mathrm{ZCR} = \frac{1}{N-1}\sum_{n=1}^{N}\left|\mathrm{sgn}(x(n)) - \mathrm{sgn}(x(n-1))\right|\,, \tag{2}$$

[4]https://huggingface.co/google/vit-base-patch16-224-in21k

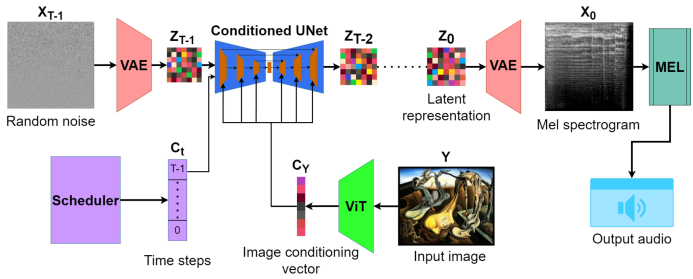


Figure 1. Pipeline of clMusDiff. The user provides an image Y, that is processed by ViT, returning a conditioning vector C_Y. To reduce the dimensionality of the data, the VAE transforms a generated random noise X_{T-1} into the initial latent representation Z_{T-1}. The denoising process starts by passing Z_{T-1}, C_Y , and time step embedding of the current time step $C_{t=T-1}$ into the Conditioned UNet denoising model. This procedure is repeated for each time step t. The last obtained representation Z_0 is decoded using VAE, resulting in the final Mel spectrogram that is transformed into the audio output by the MEL component.

where sgn is the signum function, N is the number of samples in the audio signal and $x(n)$ is the value of the audio signal at sample n.

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x(n)|^2}\,, \tag{3}$$

where N is the number of samples in the audio signal and $x(n)$ is the value of the audio signal at sample n.

These metrics were measured on all audio samples from EMID dataset, averaged and used as a baseline for comparison. Subsequently, the same metrics were computed on 65 samples produced by our model based using images from WikiArt[5]. The results were averaged, yielding SF of 0.018 (± 0.033) and 0.008 (± 0.012) for the baseline and generated samples, respectively. Baseline's ZCR of 0.080 (± 0.092) and 0.027 (± 0.027) for ours, and RMS of 0.134 (± 0.092) for the baseline and 0.152 (± 0.030) for generated.

The lower SF value may suggest that the generated audio had a more tonal or harmonic character than the other. The lower ZCR value may indicate smoother

[5]https://www.wikiart.org/

transitions in generated samples. A higher RMS than the baseline may imply increased amplitude or volume in generated samples. The obtained results indicate that the model demonstrates the capability to generate audio of quality resembling that of the baseline.

Subjective Analysis While objective metrics offer insight into audio quality, they inadequately determine if listeners perceive the music as subjectively "good" or "bad," nor properly assess the compatibility of emotions between the image and the generated audio. To assess those qualities, we conducted an anonymous survey of blind tests on 10 images and 10 corresponding audio samples generated by our model. The participants had to rate the audio quality (in the scoring range of 1 to 5) and assign one of 7 emotions to images and sounds. The participants were not informed which audio had been generated with which input image. The survey was conducted with 60 volunteers, with 34 being male and 26 female. The median age of the volunteers was 22 years. The declared backgrounds were Technical – 32, Audio – 11, Medical – 7, and Other – 10.

The results demonstrated that 6 out of 10 audio samples evoked similar emotions to their corresponding images (i.e., the most chosen emotion was the same). Some of these samples scored similar distribution of votes between emotions (see Fig. 2), which led to the conclusion that the generated audio samples can faithfully reproduce the emotional appeal of corresponding images. Another result was the switch between Sadness and Excitement in 2 samples. This could indicate that the model misinterpreted features such as dynamic expression and color variety and, as a result, generated audio with different emotional overtones.

Mean Opinion Score (MOS) of the generated audio samples was 3.407, with a standard deviation of 1.155, which corresponds to quality between ”fair” and ”good”. The analysis showed that the volunteers with medical (2.86) and audio (3.22) backgrounds knowledge assigned lower quality scores when compared to those with tech or other backgrounds, yet votes of both of those groups were the least unanimous as they show the highest standard deviation (1.23 and 1.35).

5. Conclusions and future work

In this paper, we researched various methods of image-based generation of music and proposed a diffusion-based model named clMusDiff with the capability to generate personalized audio fragments, which resemble emotions evoked by

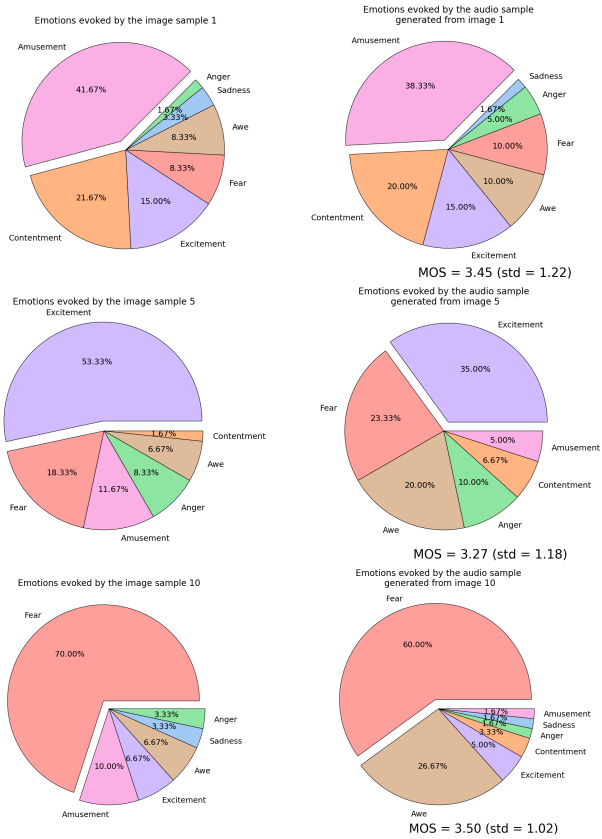


Figure 2. Subjective evaluation of example outputs. Distribution of votes between emotions attributed to the image (left) and the generated audio sample (right). The MOS and the standard deviation were provided in the lower right corner or each sample.

user-provided images. A notable application of this technique allows music therapists to incorporate unique audio content for each patient in their sessions. The evaluation showed that SF, ZCR, and RMS metrics are similar in the baseline and generated audio. Moreover, a subjective analysis of 60 volunteers showed a MOS of the generated audio equal to 3.407(±1.155) and a high correlation between the

emotions invoked by the image and the resulting audio. However, considering the subjective nature of individual music perception, the produced audio could be used for therapeutic purposes only when supervised by a trained professional.

In future work, we plan to investigate more advanced methods of spectrogram to audio conversion, e.g., self-supervised models proved effective in other branches of audio analysis. Refinement of post-processing techniques could contribute to overall quality enhancement. Moreover, creating a dataset that combines big, separate audio and image databases through emotions would be beneficial.

References

[1] Kidron, E., Schechner, Y. Y., and Elad, M. Pixels that sound. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 88–95. 2005. doi:10.1109/CVPR.2005.274.

[2] Arandjelović, R. and Zisserman, A. Objects that sound. In *Computer Vision - ECCV 2018*, pages 451–466. 2018.

[3] Xiong, Z., Lin, P.-C., and Farjudian, A. Retaining semantics in image to music conversion. In *2022 IEEE International Symposium on Multimedia (ISM)*, pages 228–235. 2022. doi:10.1109/ISM55400.2022.00051.

[4] Huang, T.-M., Hsieh, H., Qin, J., Liu, H.-F., and Eirinaki, M. Play it again IMuCo! music composition to match your mood. pages 9–16. 2020.

[5] Ruzafa, E. R. Pix2Pitch: Generating music from paintings by using conditionals GANs. *Polytechnic University of Madrid Master's Thesis*, 2020. URL `https://oa.upm.es/id/eprint/63694/contents`.

[6] Li, K., Weng, L., and Wang, X. The state of music therapy studies in the past 20 years: A bibliometric analysis. *Frontiers in Psychology*, 12, 2021.

[7] Stegemann, T., Geretsegger, M., Quoc, E. P., Riedl, H., and Smetana, M. Music therapy and other Music-Based interventions in pediatric health care: An overview. *Medicines (Basel)*, 6:25, 2019. doi:10.3390/medicines6010025.

[8] Zou, J., Mei, J., Ye, G., Huai, T., Shen, Q., and Dong, D. EMID: An emotional aligned dataset in audio-visual modality. In *Proceedings of the 1st International Workshop on Multimedia Content Generation and Evaluation: New Methods and Practice*, McGE '23, pages 41–48. 2023.

Deep Generative Models for Proton Zero Degree Calorimeter Simulations in ALICE, CERN

Patryk Będkowski[1], **Jan Dubiński**[1,2], **Kamil Deja**[1,2], **Przemysław Rokita**[1]

[1] *Warsaw University of Technology*
[2] *IDEAS NCBR*

Abstract.

Simulating detector responses is a crucial part of understanding the inner-workings of particle collisions in the Large Hadron Collider at CERN. The current reliance on statistical Monte-Carlo simulations strains CERN's computational grid, underscoring the urgency for more efficient alternatives. Addressing these challenges, recent proposals advocate for generative machine learning methods. In this study, we present an innovative deep learning simulation approach tailored for the proton Zero Degree Calorimeter in the ALICE experiment. Leveraging a Generative Adversarial Network model with Selective Diversity Increase loss, we directly simulate calorimeter responses. To enhance its capabilities in modeling a broad range of calorimeter response intensities, we expand the SDI-GAN architecture with additional regularization. Moreover, to improve the spatial fidelity of the generated data, we introduce an auxiliary regressor network. Our method offers a significant speedup when comparing to the traditional Monte-Carlo-based approaches.

Keywords: *Generative machine learning, GAN, HEP, ALICE, CERN*

1. Introduction

ALICE (A Large Ion Collider Experiment) is one of the four major detectors located at the Large Hadron Collider at CERN. One of its main goals is to replicate and study the intense conditions that existed in the early universe shortly after the Big Bang. Apart from gathering real data from the experiment, in order to understand the properties and events of particle collisions scientists have to

DOI:10.17388/WUT.2024.0002.MiNI

perform complex simulations that compared with experimental data can validate hypotheses. Such simulations are computationally extremely expensive. Existing approaches utilize statistical Monte-Carlo methods to model the physical interactions between particles. While these methods yield high-fidelity outcomes, they are also associated with high computational demands. In 2023, over 540'000 CPU devices [1] were engaged in the computations of ALICE experiments, marking a demand for developing more efficient simulation methods.

One of the most computationally expensive part of the process is the simulation of Calorimeter. Therefore, in this study, we focus on machine learning models to simulate data from the Proton Zero Degree Calorimeter (ZDC) of the ALICE experiment at CERN. We implement a deep convolutional Generative Adversarial Network (GAN) [2] as a baseline model and adapt the SDI-GAN model [3] which incorporates a regularization technique aimed at increasing the diversity of generated samples. Then, we extend the model by a simple, yet powerful regularization method focused on minimizing the difference in intensities between real and generated calorimeter responses, which improves the quality of the simulations. Finally, we introduce an auxiliary regressor which increases the model capabilities to learn accurate spatial features of the simulation data. We can summarise the contributions of this work as follows:

- We develop the first generative simulation method for the proton Zero Degree Calorimeter in the ALICE experiment at CERN.
- We evaluate the baseline GAN and SDI-GAN in our simulation task.
- We extend the SDI-GAN model by an intensity regularization loss and spatial auxiliary regressor, achieving increased simulation quality.

2. Related work

Throughout recent years, the use of Generative AI for various CERN simulations has shown the versatility of these methods [4, 5, 3, 6]. Authors of [6] employ generative machine learning algorithms for the task of simulating a Neutron ZDC calorimeter device. They propose a solution that utilizes generative models, specifically focusing on the performance of variational autoencoders and generative adversarial networks. By expanding the GAN architecture with an additional regularization, the authors significantly increase the simulation speed by two orders of magnitude while maintaining the high fidelity of the simulation.

To increase the diversity of the simulations present in the dataset, in [3] authors present a model dubbed SDI-GAN which offers significant improvements to the simulations by enforcing sample diversity among subsets of conditional data without affecting samples that exhibit consistent responses.

Finally, in order to improve how the generator learns the geometric properties of the data, in [7, 8] authors employ a regressor to accurately align the shower's center in calorimeter responses. In this work, we introduce a similar approach in the ZDC Calorimeter in ALICE.

3. Method

We select a Deep Convolutional Generative Adversarial Network [2] as our experimental model, consisting of a generator $G(z)$ and a discriminator $D(x)$. The generator creates image x from random noise z, and the discriminator differentiates between real and generated images. Both networks are conditioned on particle data and trained adversarially. Post-training, the generator is used to synthesize new calorimeter responses to particle collisions. During training, given conditional input c and k-dimensional latent code $z \sim \mathcal{N}_k(0, 1)$, generator $G(z, c)$ produces an output image $\hat{x} = G(z, c)$.

$$L_{adv}(G, D) = \mathbb{E}_{x\sim\mathcal{X},c\sim\mathcal{C}}[\log D(x, c)] + \mathbb{E}_{c\sim\mathcal{C},z\sim\mathcal{N}(0,1)}[\log(1 - D(G(c, z), c))] \quad (1)$$

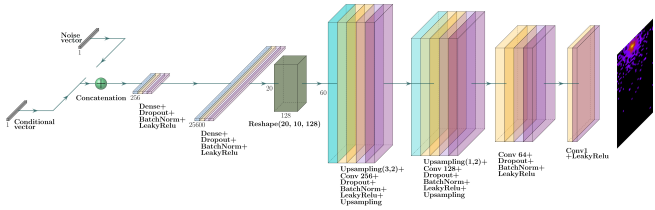


Figure 1. Architecture of the generator in used convolutional GAN across all tests.

3.1. Auxiliary regressor

One of the main properties of calorimeter response is the localisation of the center of the shower – area where the pixel values are the highest. Therefore, to

improve the GAN's learning of geometric data properties, we introduce an auxiliary regressor that trains alongside the main network to identify the 2D coordinates of the center of the collision. In preprocessing, we calculate these coordinates for all training samples, using them as targets for training. The auxiliary loss, measured by the mean squared error between the predicted $(\hat{k}_i, \hat{l}_i)$ and actual coordinates (k_i, l_i) of the highest-value pixel of the image x_i, is added to the generator's loss to refine its geometric accuracy. Strength is controlled by λ_{aux} parameter.

$$L_{aux} = \frac{1}{N} \sum_{i=1}^{N} \left[(\hat{k}_i - k_i)^2 + (\hat{l}_i - l_i)^2 \right] \tag{2}$$

3.2. Diversity regularization

Authors of SDI-GAN propose a regularization which seeks to minimize the ratio between the $L1$ distance of two images d_I generated from two distinct latent codes z_1, z_2 under the same conditioning vector c and the $L1$ distance between the latent codes themselves d_z. The measure of diversity is based on the variance in the original dataset. Thus, as a preprocessing for each unique conditioning value c, they calculate the variance of pixel values among samples. Later obtained values of all sample diversity are normalized to $[0, 1]$ by dividing by the length of dataset. This diversity measure is then multiplied with the regularization term λ_{div}, adjusting its influence on training objectives.

$$L_{div} = \sum_{i,j} \sqrt{\frac{\sum_t (x_{ij}^t - \mu_{ij})^2}{|X|}} * \left(\frac{d_I(G(c, z_1), G(c, z_2))}{d_z(z_1, z_2)} \right)^{-1} \tag{3}$$

where i and j are the pixel coordinates, t is the index of sample $x \in \chi$ and μ_{ij} is a mean value for a pixel ij.

3.3. Intensity regularization

SDI-GAN performs well on filtered data but struggles with varying intensity levels of Cherenkov light particles across distributions. To address this, we introduce a regularization using intensity measure f_{in} from the original dataset. In preprocessing, for each conditional vector c, we calculate this intensity measure as the pixel sum of the corresponding image x from χ $f_{in}(x) = \sum_{i,j} x_{ij}$, where i, j are coordinates of an image x. The intensity between the generated image $\hat{x}$ and the

original sample x corresponding to c is calculated using the mean absolute error (MAE). This loss is then multiplied by the constant λ_{in} to adjust its strength.

$$L_{in} = |f_{in}(x_c) - f_{in}(\hat{x}_c)| \tag{4}$$

3.4. Final training objective

The final training loss for our approach is a combination of the basic GAN training loss with the proposed modifications, and can be summarised as follows:

$$L(G, D) = L_{adv}(G, D) + \lambda_{div} L_{div}(G) + \lambda_{in} L_{in}(G) + \lambda_{aux} L_{aux} \tag{5}$$

4. Experiments

The Zero Degree Calorimeter (ZDC) includes a Proton (ZP) device for recording energy from non-interacting nuclei in collisions. It uses silica fibers to detect Cherenkov light, converting it into 2D images with a resolution of 56 by 30 pixels. The visualisation of the original detector responses is presented in Fig. 2 (top row). Each simulation example is therefore an image coming from a ZP device which is referred to as a response of the experiment associated with a vector of variables, referred to as conditional data. Conditional data compromises of 9 variables: energy, mass, charge, three spatial position coordinates, and three momentum coordinates.

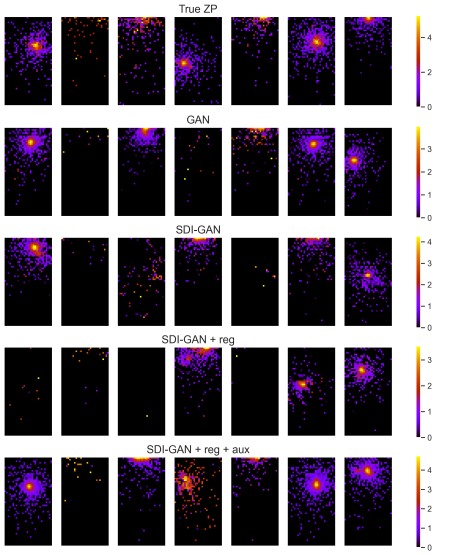


Figure 2. Example of simulated responses from different methods.

Table 1. Comparison of mean WS metric across five runs.

Model	WS Distance ↓	Std Dev
GAN	2.4752	1.6843
SDI-GAN	2.3571	1.6000
SDI-GAN + intensity reg.	2.2916	1.8210
SDI-GAN + intensity reg. + aux. reg.	**2.0777**	1.6381

The dataset employed in this experiment constitutes almost 350 thousand samples, with the validation method incorporating an 80:20 train-test split ratio. Our evaluation methodology is founded on the analysis of five distinct channels outlined in the calorimeter's specifications [9]. We employ the standard 1st Wasserstein distance metric [10] to assess the fidelity of the simulations across all channels. As shown in Tab. 1, our modifications positively influence the quality of generations as measured by the Wasserstein distance.

To visualise the differences between different methods, we plot responses to several different parameters in Fig. 2. The basic GAN produces images that are not visually aligned with the actual data. Samples generated by SDI-GAN struggle to fit the positions of real samples. Auxiliary regressor better aligns the positions of centers of particle showers. In Fig. 4 we present the visual results for channel 4 and 5, as they contain most of information.

5. Conclusions

In this study, we applied and extended generative machine learning models to simulate the intricate dynamics of the Proton Zero Degree Calorimeter in the ALICE experiment at CERN. By establishing GAN as a baseline model we tested its performance in comparison to the SDI-GAN. Furthermore, we tailored SDI-GAN to the unique demands of High Energy Physics experiments, by adding additional regularization on the sum of pixel intensities which proved to increase the fidelity of the simulations. Further incorporation of auxiliary regressor proved essential to securing the lowest WS results.

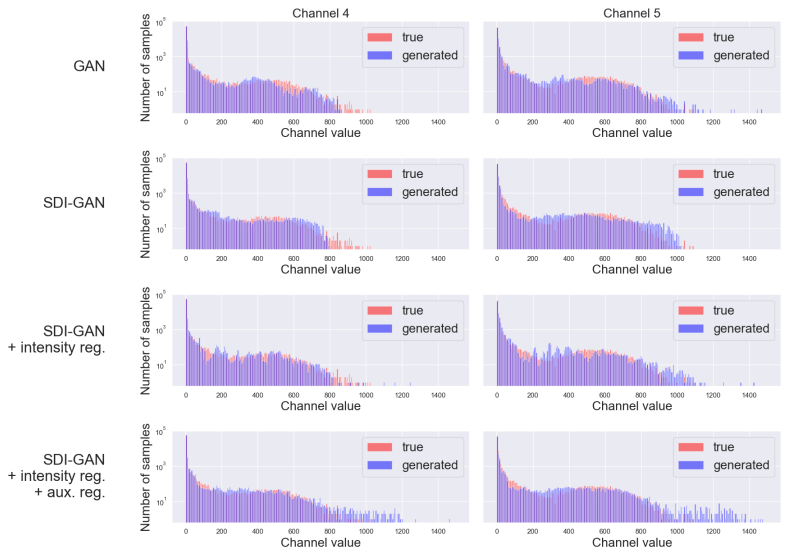


Figure 3. Histograms of true and generated distributions of channel values. The GAN and SDI-GAN model have visible problems with underproducing high-energy responses. The implementation of additional regularization, and auxiliary regressor positively influence better alignment to true distribution, but tends to oversample the high-energy responses.

Appendix: training details

Fine-tuning was conducted via grid search across logarithmically spaced values: $\lambda_{div} \in \{10^{-2}, 10^{-1}, 10^{0}\}$, $\lambda_{in} \in \{10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}\}$, and $\lambda_{aux} \in \{10^{-4}, 10^{-3}, 10^{-2}\}$. Model configurations were assessed for the lowest WS over five runs, with the best performance yielded by $\lambda_{div} = 10^{-1}$, $\lambda_{in} = 10^{-10}$, and $\lambda_{aux} = 10^{-3}$. The rest of the parameters were constant for all tests.

6. Acknowledgments

This research was funded by National Science Centre, Poland grants: 2020/39/O/ST6/01478 and 2022/45/B/ST6/02817. This research was supported in part by PLGrid Infrastructure grants: PLG/2023/016393, PLG/2023/016361, PLG/2023/016278.

References

[1] Worldwide LHC Computing Grid Resources Report. Remote access (16.01.2024): http://wlcg-cric.cern.ch/core/vopledgereq/listcomp/, 2024.

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., and Xu, e. a. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[3] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Selectively increasing the diversity of gan-generated samples. In *Neural Information Processing*. 2023. doi:10.1007/978-3-031-30105-6_22.

[4] Deja, K., Trzciński, T., Graczykowski, Ł., and Collaboration, A. Generative models for fast cluster simulations in the tpc for the alice experiment. In *Conference on Information Technology, Systems Research and Computational Physics*, pages 267–280. Springer, 2018.

[5] Deja, K., Dubiński, J., Nowak, P., Wenzel, S., and Trzciński, T. End-to-end sinkhorn autoencoder with noise generator, 2020.

[6] Jan Dubiński, S. W. e. a., Kamil Deja. Machine learning methods for simulating particle response in the zero degree calorimeter at the alice experiment, cern. abs/2306.13606, 2023. URL https://api.semanticscholar.org/CorpusID:259243732.

[7] Chekalina, V., Orlova, E., Ratnikov, F., Ulyanov, D., Ustyuzhanin, A., and Zakharov, E. Generative Models for Fast Calorimeter Simulation: the LHCb case. *EPJ Web Conf.*, 2019. doi:10.1051/epjconf/201921402034.

[8] Alexander Rogachev, F. R. Gan with an auxiliary regressor for the fast simulation of the electromagnetic calorimeter response, 2022. doi:10.48550/arXiv.2207.06329.

[9] Dellacasa, G. e. a. ALICE technical design report of the zero degree calorimeter (ZDC). 1999.

[10] Ilya Tolstikhin, S. G. e. a., Olivier Bousquet. Wasserstein auto-encoders, 2017. doi:10.48550/arXiv.1711.01558.

Particle physics DL-simulation with control over generated data properties

Karol Rogoziński[1], **Jan Dubiński**[1,2], **Przemysław Rokita**[1], **Kamil Deja**[1,2]

[1] *Warsaw University of Technology*
[2] *IDEAS NCBR*
rogo.karol@gmail.com

Abstract. *The research of innovative methods aimed at reducing costs and shortening the time needed for simulation, going beyond conventional approaches based on Monte Carlo methods, has been sparked by the development of collision simulations at the Large Hadron Collider at CERN. Deep learning generative methods including VAE, GANs and diffusion models have been used for this purpose. Although they are much faster and simpler than standard approaches, they do not always keep high fidelity of the simulated data. This work aims to mitigate this issue, by providing an alternative solution to currently employed algorithms by introducing the mechanism of control over the generated data properties. To achieve this, we extend the recently introduced CorrVAE, which enables user-defined parameter manipulation of the generated output. We adapt the model to the problem of particle physics simulation. The proposed solution achieved promising results, demonstrating control over the parameters of the generated output and constituting an alternative for simulating the ZDC calorimeter in the ALICE experiment at CERN.*
Keywords: *generative machine learning, high energy physics, ALICE, CERN*

1. Introduction

The Large Hadron Collider is the most important instrument at CERN. It is the world's most powerful particle accelerator and is part of the CERN accelerator complex. Particle beams collide at four locations around the LHC, corresponding to the four main detectors: the LHCb [1], ATLAS [2], CMS [3] and ALICE [4].

DOI:10.17388/WUT.2024.0002.MiNI

In this work, we focus on the latter, and in particular its Zero Degree Neutron Calorimeter (ZDC) [5], which collects the energy of observers, i.e., particles that did not participate directly in the collision. It is made of optical fibers arranged in a 44x44 grid. From a physicist's perspective, simulations of calorimeter responses are just as crucial as experiments. If the results of simulations align with actual data, they can be considered as the real course of physical events and used for scientific research and analysis. Traditional methods of conducting simulations are costly and time-consuming [6], so researchers have been developing alternative, more efficient methods for simulating collisions. This is further described in [7, 8].

The objective of this work is to develop a machine learning model for rapid simulation of the ZDC calorimeter in the ALICE experiment, while also enabling control over the properties of the generated data. To achieve this, we utilized the CorrVAE model proposed in [9]. The method was initially designed to create images based on user-defined parameters, such as color, shape, or object position, and showed promising results. In this paper, we propose to adapt it to the high-energy physics (HEP) scenario. To that end, we introduce a new architecture that aligns with the ZDC response requirements, and introduce additional encoder with conditional latent space to allow conditional generations from particle properties.

In our experiments we show that our approach offers an alternative to current simulation methods, with a particular focus on controlling the response properties of particles, as presented in [9]. By assuming the physical parameters of the particles, the model is able to generate calorimeter responses that are both realistic and modifiable in a predetermined manner.

2. Related Work

To address the issues of the computational complexity of HEP simulations, researchers actively explore the possiblity of using generative machine learning methods instead. In [10], the authors simulated an electromagnetic calorimeter from the LHCb detector using the Wasserstein GAN [11] model. To better represent momentum and position, the authors utilized a generator, discriminator, and an auxiliary regressor network to evaluate consistency. A similar idea was also presented in [12], where the authors also used additional regressors for measuring the energy and position of the collisions.

The simulations of the Zero Degree Calorimeter Regarding the ZDC calorimeter of the ALICE experiment, in [8], authors introduced a method that based on Sinkhorn's autoencoder architecture [13]. It treats the calorimeter simulation problem as generating an image with the dimensions of a neutron calorimeter ZDC, which is 44x44. [8] replaces the classical regularization in the latent space of VAE models with an additional noise generator defined as a neural network. The latest work on the subject is [14], which presents two new approaches to solving the problem. The first method is based on Conditional VAE [15]. The second method, based on Conditional GAN [16], also incorporates conditional information into the learning and generation process. Additionally, similar to the approach in [10], the authors utilized an auxiliary regressor to verify the accuracy of the particle's position. Another work that focuses on simulating the ZDC calorimeter [7] employs a GAN network with added selective diversity increase loss regularization.

Controlling properties in generative modeling Generating data with specific properties has been the subject of much work [17, 18, 19, 9, 20, 21]. The main concept behind those work is to combine variational autoencoders [17, 20] with various techniques such as graph networks [18, 19]. However, all the works mentioned have the problem of generating examples with given properties that are correlated with each other. This problem is solved by [9], which served as the basis in the development of this work.

3. Problem Formulation

The task is to generate simulation responses based on information about the particle at the time of the collision. To that end, we use a dataset described in [22], which includes responses generated by classical Monte Carlo methods. The dataset contains nearly 300,000 simulated responses, each with 44x44 dimension, and includes nine properties for each particle: momentum and velocity in three planes, as well as mass, charge, and energy.

4. Method

The CorrVAE model was used as a starting point for this work. Its structure is based on the classical VAE, but it introduces two latent spaces instead of one. The first, w, is responsible for the image properties that the user has indicated, such

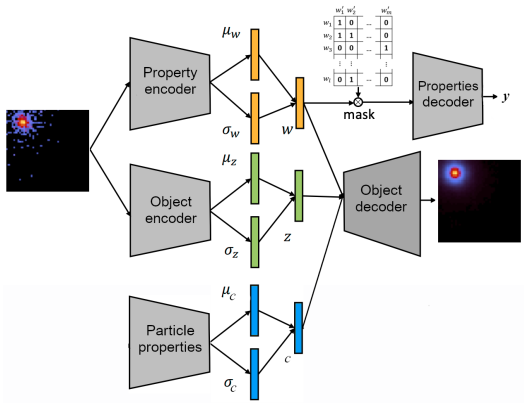


Figure 1: Model structure presentation. Unlike the classic CorrVAE, the information is encoded in three latent spaces, where the additional encoder encodes particle properties.

as size and colour. The second latent space, z, is responsible for all other image features. These spaces are independent of each other, and the information they encode does not intersect in any way. Additionally, the authors proposed a mask with the same number of rows as the user-defined parameters and the same number of columns as the hidden space elements. A value of 1 in their intersection indicates that the element encodes a specific property. During the training phase, the property decoder receives the elements of the w latent space multiplied by the mask. The decoder's output is then compared to the original user-defined properties using MSE as an additional loss function.

Basic adaptation for ZDC simulation To adapt the CorrVAE model for the simulation problem, several modifications were made. These changes were necessary due to the different specificity of the data compared to what was assumed in the original model. The modifications included changing the activation function in the last layer from Sigmoid to ReLU and replacing the reconstruction loss function from Binary Cross Entropy (BCE) to Mean Squared Error (MSE) due to the wider range of data generated by the simulation. The convolutional layers had to be adjusted to a size of 44x44, which is unusual for image generation problems.

Model modifications Additionally, we modify the baseline model by adding conditioning to generate responses from the particle data in real-world conditions. However, a straight-forward conditioning is not compatible with the CorrVAE model, because the model assumes the independence of the hidden spaces and the ability to control the defined parameters with the hidden space w. Therefore, an additional encoder was added along with the hidden space c. The input of the encoder was the particle data, and the output stored information about the values and size of the response. The hidden space w was only responsible for its position. New responses were generated by inputting data about the particle into the vector c to encode its physical properties, optimizing the vector w with a property decoder to include information about its position, and sampling the vector z from a normal distribution. The final model structure is shown in Figure 1.

5. Experiments

In this section, we provide the experimental evaluation of our method. Beside the standard evaluation of image reconstruction quality, the ZDC calorimeter simulation presents an unconventional case where we can directly measure the physical properties of the generated outputs. To that end, we follow [8], and calculate values for five different channels that are used for physical analysis. We compare the distribution of those channels between generated and true images, with the Wasserstain distance, and average it across all channels.

We compare the results of different simulation approaches in Tab. 1. Additionally in Fig. 2 we show that our method yields much better alignment to the original responses (top row), when analysing the position of the center of the collision. However, similarly to other VAE-based models, we can see slightly blurred responses, which is the opposite of what happens in models based on the GAN architecture.

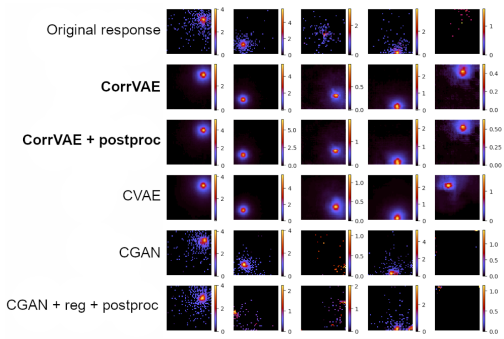


Figure 2: Comparison of randomly selected simulations generated by different models.

	MSE	Wasserstein
CorrVAE	**1.03**	**16.15**
CorrVAE + postproc	**1.18**	**3.83**
CVAE	1.02	6.35
CGAN	2.96	8.27
CGAN + reg + postproc	2.98	5.15

Table 1: Comparison of results for the HEP dataset. Initially weaker than the other adapted models, the adapted CorrVAE model achievs the best results after applying post-processing. Values averaged over 3 runs.

Controlling the properties To show the control over the properties of generated responses with our method, in the final experiment we show how we can adjust the position of the center of the mass, by interpolating between the selected values of the latent space. In particular, we train the model in a way that only selected two dimensions of the latent space w are correlated with the position of the center of the mass (w_1 for x position and w_2 for y position). In Fig. 3, we show the effect of traversing through those dimensions.

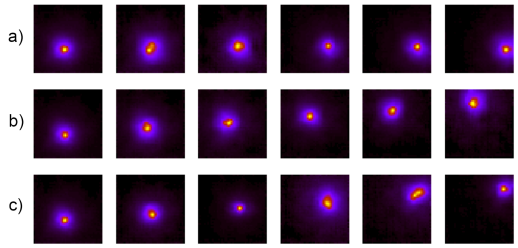


Figure 3: Generated images of presented model by traversing two latent variables in w for HEP dataset according to the mask between x, y position and eight-element w vector. (a) Traversing on the w_1 that controls x position; (b) Traversing on the w_2 that controls y position; (c) Traversing on the w_1 and w_2 at the same time.

6. Conclusions

In the paper, we successfully adapt and develop the CorrVAE model for high-energy physics simulation. This includes introduction of a new architecture, with additional conditional space needed to generate new simulations from the particle

data. Our approach enables control over individual properties, which could be manipulated by feeding synthetic parameters or using the original particle properties at the time of the inference. The model prepared in this way can compete with the best methods currently in use.

Acknowledgments

This research was funded by National Science Centre, Poland grants: 2020/39/O/ST6/01478 and 2022/45/B/ST6/02817. This research was supported in part by PLGrid Infrastructure grants: PLG/2023/016393, PLG/2023/016361, PLG/2023/016278.

References

[1] Alves Jr, A. A., Andrade Filho, L., Barbosa, A., Bediaga, I., Cernicchiaro, G., Guerrer, G., Lima Jr, H., Machado, A., Magnin, J., Marujo, F., et al. The lhcb detector at the lhc. *Journal of instrumentation*, 3(08):S08005, 2008.

[2] Collaboration, A. The atlas experiment at the cern large hadron collider. *Journal of Instrumentation*, 3, 2008.

[3] Collaboration, C. The cms experiment at the cern lhc. *Journal of Instrumentation*, 3, 2008.

[4] Collaboration, A. The alice experiment at the cern lhc. *Journal of Instrumentation*, 3, 2008.

[5] M. Galio, W. K. and Lestam, L. Alice zero-degree calorimeter (zdc) : Technical design report. *Technical design report. ALICE*, 3, 1999.

[6] Paganini, M., de Oliveira, L., and Nachman, B. CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Phys. Rev. D*, 97(1):014021, 2018. doi:10.1103/PhysRevD.97.014021.

[7] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Selectively increasing the diversity of gan-generated samples, 2023.

[8] Deja, K., Dubiński, J., Nowak, P., Wenzel, S., and Trzciński, T. End-to-end sinkhorn autoencoder with noise generator, 2020.

[9] Wang, S., Guo, X., Lin, X., Pan, B., Du, Y., Wang, Y., Ye, Y., Petersen, A. A., Leitgeb, A., AlKhalifa, S., Minbiole, K., Wuest, W., Shehu, A., and Zhao, L. Multi-objective deep data generation with correlated property control, 2022.

[10] Chekalina, V., Orlova, E., Ratnikov, F., Ulyanov, D., Ustyuzhanin, A., and Zakharov, E. Generative models for fast calorimeter simulation: the lhcb casegt;. *EPJ Web of Conferences*, 214:02034, 2019. ISSN 2100-014X. doi:10.1051/epjconf/201921402034. URL http://dx.doi.org/10.1051/epjconf/201921402034.

[11] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan, 2017.

[12] Erdmann, M., Glombitza, J., Quast, and Thorben. Precise simulation of electromagnetic calorimeter showers using a wasserstein generative adversarial network. *Computing and Software for Big Science*, 3(1), 2019. ISSN 2510-2044. doi:10.1007/s41781-018-0019-7. URL http://dx.doi.org/10.1007/s41781-018-0019-7.

[13] Patrini, G., van den Berg, R., Forré, P., Carioni, M., Bhargav, S., Welling, M., Genewein, T., and Nielsen, F. Sinkhorn autoencoders, 2019.

[14] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Machine Learning methods for simulating particle response in the Zero Degree Calorimeter at the ALICE experiment, CERN. Technical report, 2023. URL https://cds.cern.ch/record/2875960.

[15] Debbagh, M. Learning structured output representations from attributes using deep conditional generative models, 2023.

[16] Mirza, M. and Osindero, S. Conditional generative adversarial nets, 2014.

[17] Klys, J., Snell, J., and Zemel, R. S. Learning latent subspaces in variational autoencoders. *CoRR*, abs/1812.06190, 2018. URL http://arxiv.org/abs/1812.06190.

[18] You, J., Liu, B., Ying, R., Pande, V. S., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *CoRR*, abs/1806.02473, 2018. URL http://arxiv.org/abs/1806.02473.

[19] Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. Graphaf: a flow-based autoregressive model for molecular graph generation. *CoRR*, abs/2001.09382, 2020. URL https://arxiv.org/abs/2001.09382.

[20] X. Guo, Y. D. and Zhao, L. Property controllable variational autoencoder via invertible mutual dependence. *International Conference on Learning Representations*, 2020.

[21] Henter, G. E., Lorenzo-Trueba, J., Wang, X., and Yamagishi, J. Deep encoder-decoder models for unsupervised learning of controllable speech synthesis, 2018.

[22] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Machine learning methods for simulating particle response in the zero degree calorimeter at the alice experiment, cern, 2023.

Generative Diffusion Models for Fast Simulations of Particle Collisions at CERN

Mikołaj Kita[1], **Jan Dubiński**[1,2], **Przemysław Rokita**[1], **Kamil Deja**[1,2]

[1] *Warsaw University of Technology*
[2] *IDEAS NCBR*
mikolajkita@gmail.com

Abstract. *In High Energy Physics simulations play a crucial role in unraveling the complexities of particle collision experiments within CERN's Large Hadron Collider. Machine learning simulation methods have garnered attention as promising alternatives to traditional approaches. While existing methods mainly employ Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs), recent advancements highlight the efficacy of diffusion models as state-of-the-art generative machine learning methods. We present the first simulation for Zero Degree Calorimeter (ZDC) at the ALICE experiment based on diffusion models, achieving the highest fidelity compared to existing baselines. We perform an analysis of trade-offs between generation times and the simulation quality. The results indicate a significant potential of latent diffusion model due to its rapid generation time.*

Keywords: *diffusion models, particle simulations, ALICE, CERN*

1. Introduction

The European Organization for Nuclear Research (CERN) is one of the world's most renowned centers for scientific research in High-Energy Physics (HEP), where the Large Hadron Collider (LHC) [1], the world's largest and most powerful particle accelerator is located. To understand what happens during these collisions, sophisticated simulations of the detectors inside the LHC are generated. The majority of these simulations are founded on Monte Carlo methods [2], which yield results of high quality, albeit with a significant computational expense. In response, researchers investigate strategies to optimize resource use, including the application of generative machine learning methods to lessen CPU consumption [3], [4].

DOI:10.17388/WUT.2024.0002.MiNI

Current efforts to develop generative simulations for ZDC include the usage of Variational Autoencoders (VAEs) [5] and Generative Adversarial Networks (GANs) [6] for simulating particle responses [3, 7, 4]. While they present an improvement over the existing methods based on Monte Carlo, several challenges have been identified, notably the production of indistinct samples in VAEs and an unsatisfactory variety of samples generated by GANs. Methods to enhance the diversity of GAN-generated samples and to diminish the blurriness of VAE-generated images exist [4, 7]. Nevertheless, a promising alternative to such approaches is the usage of the newly proposed diffusion models. Originally, diffusion models were employed in image generation [8] and have since branched out to other areas like video [9], text-to-audio [10], and even text generation [11]. The demonstrated versatility of diffusion models presents potential for their application in various fields [12, 13, 14], including physics simulations. Therefore, in this work, we introduce a new simulation method for ZDC that leverages diffusion models. Our method achieves better performance on Wasserstein distance evaluation metric for conditional diffusion model and presents significant potential of latent diffusion model due to its rapid generation time.

2. Related work

Numerous efforts have been directed towards supplanting the resource-intensive Monte Carlo methods traditionally used for High Energy Physics simulations with more efficient alternatives like generative modeling. Most of the proposed solutions are based on GANs, such as [3, 7, 15], but other methods such as generative autoencoders [4] or score-based generative models are also used [16]. The proliferation of those approaches lead to the creation of Fast Calorimeter Simulation Challenge [17] with three high-quality datasets with calorimeter responses to particle collisions.

Apart from calorimeters simulations, diffusion models find extensive application in physics simulations. Their utilization spans from the reconstruction of flow fields [18] to the generation of jets within the realm of HEP [19, 20].

3. Zero Degree Calorimeter simulation

The neutron Zero Degree Calorimeter (ZDC) is a quartz-fiber spaghetti calorimeter, utilizing the principle of detecting Cherenkov light produced by charged par-

ticles of the shower in silica optical fibers [21]. Every alternate fiber is directed towards a photomultiplier (PMTc), with the rest of the fibers being grouped into bundles that lead to four distinct photomultipliers (PMT1 to PMT4). The design allows for precise measurement of neutron energy in heavy ion collisions at the CERN LHC.

The response of the ZDC is treated as a one-channel image composed of a 44 by 44 pixel matrix, wherein the value of each pixel represents the count of photons deposited in an individual fibre, which we visualize in Fig. 1.

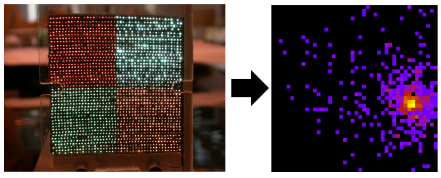


Figure 1. ZDC cross-section with visible optic fibres grid (left) represented as a one-channel image (right)

Every response is produced by a particle, which is characterized by attributes: mass, energy, charge, momenta, and primary vertex. The correlation between particle attributes and responses is non-deterministic. Consequently, a single set of nine attributes may yield a variety of different responses, necessitating evaluations across a distribution of generated images. The dataset consists of 300000 image-particle pairs from GEANT4 simulation tool.

4. Method

Diffusion models consist of *forward diffusion pass* $q(x_{1:T}|x_0)$, which is described by a Markov chain that progressively adds noise to the original data, and a *reverse diffusion pass* $p_\theta(x_{0:T})$, which aims to reconstruct the original image from the noised data with a trainable model, as referenced in Fig. 2.

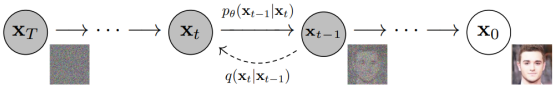


Figure 2. Graphical presentation of forward and reverse diffusion pass. Image from [8]

An alternative and faster method to sample images from diffusion models is proposed in [22]. The authors reinterpret the reverse diffusion process as a non-Markovian process, which allows for omission of intermediate steps in the denoising proccess, enabling faster generation of samples as presented in Fig. 3

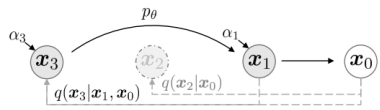


Figure 3. Accelerated generation with DDIM sampling method. Image from [22]

We use a diffusion model based on UNet architecture [23]. To introduce conditioning on the particle parameters, we adapt the text-conditioning of the text-to-image diffusion models. Particle characteristics y are multiplied by a linear layer $\tau_\theta(y)$ to function as an embedding for physical conditions. This allows the model to focus on specific features of the generated data based on the particle data input, enhancing contextual relationships between them:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \tag{1}$$

where

$$Q = W_Q^{(i)} \cdot \phi_i(z_t), \quad K = W_K^{(i)} \cdot \tau_\theta(y), \quad V = W_V^{(i)} \cdot \tau_\theta(y). \tag{2}$$

where $W_Q^{(i)}$, $W_K^{(i)}$, and $W_V^{(i)}$ are learnable matrix projections for query, key and value [24].

During additional experiments, we also investigate Latent Diffusion Models [24], which run the diffusion process in the latent space of an image autoencoder.

5. Experiments and Results

5.1. Evaluation

The evaluation process unfolds in two phases. Initially, a test dataset comprising 60,000 images from GEANT4 simulation tool is utilized. Five channel values, corresponding to PMT1 to PMT4 and PMTc, are extracted from each image. Subsequently, a diffusion model generates a comparable set of 60,000 images. The same procedure for channel extraction is applied to these images. The critical step involves calculating the disparity between each pair of corresponding channel distributions using the Wasserstein Distance, yielding five distinct distances. The final step in the evaluation is to compute the average of these five distances. The primary goal is to attain the lowest possible average.

5.2. Results

Results are based on generating 60000 images for two sampling methods: DDPM [8] and DDIM [22] with baseline methods from [7]. When comparing sampling methods, the DDIM method shows a slight advantage at lower inference steps, while the DDPM method notably excels at the higher end of the spectrum.

Table 1. Conditional Diffusion Model results based on different sampling methods and number of inference steps, evaluated on a single GeForce RTX 3090 GPU.

Method	Inference steps	Mean WS	ch1	ch2	ch3	ch4	ch5	Time (minutes)
GAN	-	8.2	4.4	5.5	7.3	9.1	15.0	<1
VAE	-	6.4	4.6	5.2	4.2	9.1	13.7	<1
end2end SAE	-	6.3	4.2	5.0	4.0	4.0	13.6	<1
DCGAN + selective div increase	-	4.5	2.2	4.0	4.4	6.2	8.0	<1
DDIM	20	9.9	5.9	5.5	6.2	6.0	25.8	6
DDIM	50	6.4	3.9	3.4	3.9	3.0	17.8	15
DDIM	100	21.6	13.9	13.2	14.0	12.7	54.3	31
DDPM	20	10.1	6.5	6.6	7.0	7.3	23.0	6
DDPM	50	7.2	4.6	4.8	4.6	5.1	16.9	15
DDPM	100	4.7	3.0	3.2	2.9	3.2	11.0	31
DDPM	250	2.1	1.5	1.5	1.4	1.6	4.5	56
DDPM	500	1.2	0.9	1.1	0.8	1.0	2.1	109
DDPM	1000	1.8	1.7	1.8	1.9	1.0	2.4	215

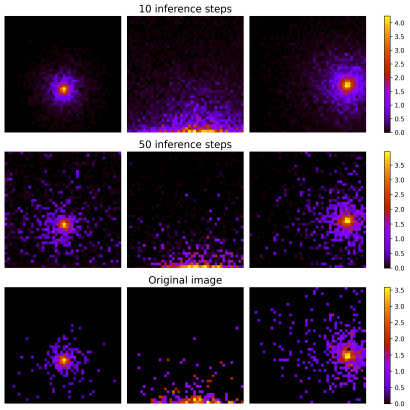


Figure 4. Conditional Diffusion Model generations from the DDIM sampler at different number of inference steps

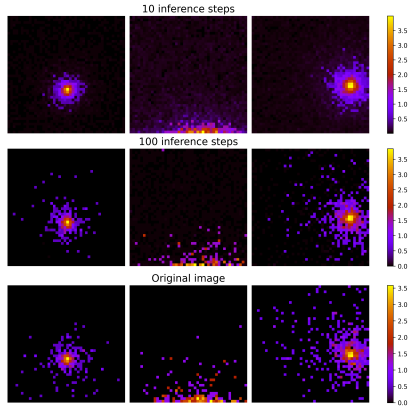


Figure 5. Conditional Diffusion Model generations from the DDPM sampler at different number of inference steps

The average Wasserstein distance diminishes up to a certain number of infer-

ence steps, before it starts to rise. This turnaround happens more swiftly with the DDIM sampler and more gradually with the DDPM sampler.

The progressive decrease in Wasserstein Distance can be seen in Fig. 4 and Fig. 5. Images that initially appear slightly blurred at 10 inference steps gradually achieve a quality that is nearly indistinguishable from the original.

5.3. Additional Experiments

The superior performance exhibited by conditional diffusion models is accompanied by the substantial cost of increased computational time. In response to this challenge, we have chosen to augment the model with the integration of a Latent Diffusion Model [24]. The core concept of LDM involves the integration of an autoencoder that compresses the high-dimensional pixel space into a more computationally manageable latent representation, thus reducing the complexity and resource demands of both training and generative process. We used a conditional VAE with scaling factor 0.18215 and latent space with dimensions [4,5,5].

Table 2. Comparison of diffusion models on Wasserstein Distance and Time performance on a GeForce RTX 3090 GPU

Model	Average WS	WS ch1	WS ch2	WS ch3	WS ch4	WS ch5	Time (minutes)
Latent Diffusion Model	12.6	5.8	8.8	9.1	10.5	28.6	<1
Conditional Diffusion Model	1.2	0.9	1.1	0.8	1.0	2.1	109

As shown in Tab. 2, the Latent Diffusion Model presents significant potential due to its rapid generation times. However, it still requires more development work to reach the performance benchmarks set by other models.

6. Conclusions

In this paper, we apply generative diffusion models to the problem of simulating the ZDC calorimeter at the ALICE experiment at CERN. We demonstrate the capabilities of conditional diffusion models, outperforming existing state-of-the-art models by achieving lower Wasserstein Distances and generating high-quality, realistic images. Moreover, we show that the Latent Diffusion Model presents significant potential due to its rapid generation times. However, it still requires more development work to reach the performance benchmarks set by other models, which we leave as future work.

Acknowledgments

This research was funded by National Science Centre, Poland grants: 2020/39/O/ST6/01478 and 2022/45/B/ST6/02817. This research was supported in part by PLGrid Infrastructure grants: PLG/2023/016393, PLG/2023/016361, PLG/2023/016278.

References

[1] Evans, L. and Bryant, P. LHC Machine. *Journal of Instrumentation*, 3:S08001, 2008.

[2] Allison, J. and et al. Recent developments in geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 2016.

[3] Michela, P., de Oliveira Luke, and Benjamin, N. Calogan: Simulating 3d high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D*, 97(1), 2018.

[4] Deja, K., Dubiński, J., Nowak, P., Wenzel, S., Spurek, P., and Trzcinski, T. End-to-end sinkhorn autoencoder with noise generator. *IEEE Access*, 9:7211–7219, 2021.

[5] Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. ICLR, 2013.

[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680. 2014.

[7] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Machine learning methods for simulating particle response in the zero degree calorimeter at the alice experiment, cern. In *AIP Conference Proceedings*, volume 3061. AIP Publishing, 2024.

[8] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*. NeurIPS, 2020.

[9] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. 2022.

[10] Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 21450–21474. PMLR, 2023.

[11] Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*. ICLR, 2023.

[12] Kaleta, J., Dall'Alba, D., Płotka, S., and Korzeniowski, P. Minimal data requirement for realistic endoscopic image generation with stable diffusion. *International journal of computer assisted radiology and surgery*, 2023.

[13] Karras, J., Holynski, A., Wang, T.-C., and Kemelmacher-Shlizerman, I. Dreampose: Fashion video synthesis with stable diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22680–22690. 2023.

[14] Liu, M. and Hu, Y. Application potential of stable diffusion in different stages of industrial design. In H. Degen and S. Ntoa, editors, *Artificial Intelligence in HCI*, pages 590–609. Springer Nature Switzerland, Cham, 2023.

[15] Dubiński, J., Deja, K., Wenzel, S., Rokita, P., and Trzciński, T. Selectively increasing the diversity of gan-generated samples, 2023.

[16] Mikuni, V. and Nachman, B. Score-based generative models for calorimeter shower simulation. *Physical Review D*, 2022.

[17] Giannelli, M. F., Gregor Kasieczka, C. K., Nachman, B., Salamani, D., Shih, D., and Zaborowska, A. Fast Calorimeter Simulation Challenge 2022 . Online access (16.01.2024): `https://calochallenge.github.io/homepage/`, 2022.

[18] Shu, D., Li, Z., and Barati Farimani, A. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023. ISSN 0021-9991.

[19] Leigh, M., Sengupta, D., Quétant, G., Raine, J. A., Zoch, K., and Golling, T. Pc-jedi: Diffusion for particle cloud generation in high energy physics. *SciPost Physics*, 16(1), 2024. ISSN 2542-4653.

[20] Mikuni, V., Nachman, B., and Pettee, M. Fast point cloud generation with diffusion models in high energy physics. *Physical Review D*, 108(3), 2023. ISSN 2470-0029.

[21] et al., R. A. The neutron zero degree calorimeter for the ALICE experiment. *Nuclear Instruments and Methods in Physics Research Section A-accelerators Spectrometers Detectors and Associated Equipment*, 564:235–242, 2006.

[22] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*. ICLR, 2020.

[23] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015.

[24] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2022.

Chapter 12

Interdisciplinary Topics in Artificial Intelligence

Track Chairs:

- prof. Adam Wojciechowski - Lodz University of Technology
- prof. Jarosław Wąs - AGH University of Krakow
- prof. Maciej Grzenda - Warsaw University of Technology

Turning feed classification using SVM

Andrzej Chmielowiec[1[0000−0001−6629−0029]],
Adam Michajłyszyn[1[0000−0002−6458−387X]],
Sylwia Sikorska-Czupryna[1[0000−0002−0063−085X]],
Leszek Klich[1[0000−0001−6099−2417]],
Paweł Przybyłowicz[2[0000−0001−7870−8605]]

[1]*Rzeszow University of Technology, Faculty of Mechanics and Technology*
Kawiatkowskiego 4, 37-450 Stalowa Wola, Poland
achmie@prz.edu.pl
[2]*AGH University of Krakow, Faculty of Applied Mathematics*
Mickiewicza 30, 30-059 Kraków, Poland
pprzybyl@agh.edu.pl

Abstract. *The article introduces an innovative feed rate classification method for the turning process. Its operation is based on the SVM method, coupled with the use of a rapid laser micrometer, enabling the entire process to be carried out in production conditions. The presented solution aligns with the trend of utilizing machine learning for automating and improving production processes.*
Keywords: *SVM, turning, feed rate, process control, micrometer*

1. Introduction

Machine learning methods based on SVM are widely used to enhance the quality and efficiency of production processes. One of such processes is the turning process, which involves rotating the workpiece around an axis and removing excess material using a cutting tool. Primarily, publications focus on monitoring the entire turning process [1], with particular emphasis on detecting tool wear [2, 3, 4]. Solutions are also proposed to optimize the selection of parameters for high-speed turning processes [5]. The SVM method is also applied to correlate the roughness of the final surface with signals from the process, such as sound [6] and vibrations [7]. In this article, the authors extend the developed statistical methods [8, 9]

DOI:10.17388/WUT.2024.0002.MiNI

for studying production systems to the Support Vector Machine (SVM) method, which they use to classify the feed rate as one of the key parameters in the turning process. A highly innovative aspect of the experiment is the use of a rapid laser micrometer, enabling full automation of the measurement process and verification of turning parameters in production conditions.

2. Materials and Methods

The main objective of the methods described in this section is to demonstrate that it is possible to accurately classify the feed rate (Figure 1) based on a series of precise measurements. The presented methods specify three parameters that influence the preparation of input data for the machine learning process. It is important to emphasize that the SVM model used is not trained by direct measurement results but rather by data subjected to prior processing. Therefore, the main focus of the article is to indicate a data processing method that would ensure the highest accuracy of the SVM model. The source data were obtained from the

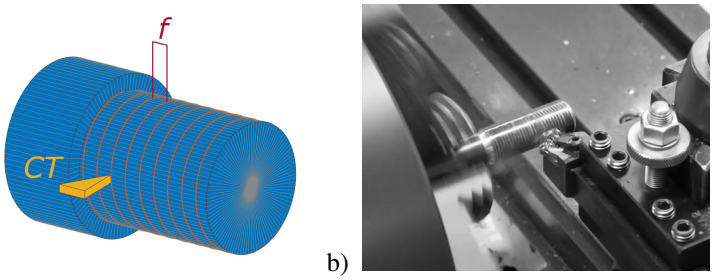


Figure 1. Turninng: a) process scheme with cutting tool (CT), its trajectory and feed rate f, b) real process photo

measurement station presented in Figure 2. Six samples made of S235 steel were examined in a turning process with feed rates ranging from 0.20 to 0.50 mm/rev at a constant cutting speed of 100 m/min (Figure 3). Each measurement allowed for obtaining approximately 25,000 values. Ultimately, 30 measurements were conducted for each sample at different angles of rotation relative to the measuring laser beam. During measurement, each sample was moved at a constant speed of 1.0 mm/s, intersecting the laser beam of the scanning micrometer LSM503. The

data were then transferred to the computer via a serial port. In total, approximately 4.5 million measurement values were collected throughout the process.

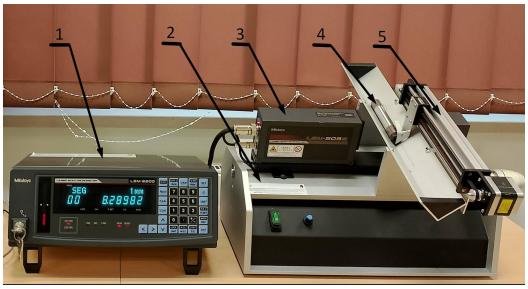


Figure 2. Measuring station: 1) Mitutoyo LSM 6200 display, 2) tilting base, 3) Mitutoyo LSM-503S laser scanning micrometer, 4) tested sample, 5) automated sample feed drive

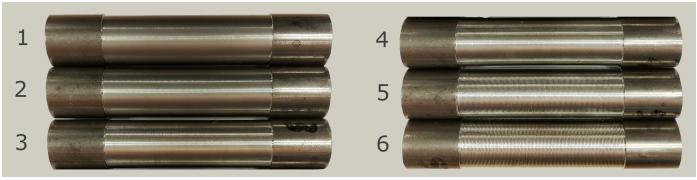


Figure 3. Tested samples: 1 (0.20 mm/rev), 2 (0.26 mm/rev), 3 (0.32 mm/rev), 4 (0.38 mm/rev), 5 (0.44 mm/rev), 6 (0.50 mm/rev)

Figure 4a) illustrates a typical reading of the single measurement. A significant deviation of the measured sample from the horizontal line can be easily observed. This effect is caused by the pressure of the cutting tool during turning. To eliminate the effect arising from the mechanical properties of the process, it was decided to determine a quadratic approximation of the measurement using the least squares method (*Square approximation* curve in Figure 4). Subsequently, the obtained function was subtracted from the measurement result, yielding values oscillating around 0. On the other hand, Figure 4 b) depicts the same data transformed in such a way that each point represents a 32-point moving average of the direct measurements preceding it, calculated as $y_k = \frac{1}{32} \sum_{i=k-31}^{k} x_i$. The figure clearly shows that this approach filters out high frequencies and significantly reduces the data

variance. In the subsequent part of the article, by MAVG 1, we will understand the moving average of length 1, which essentially represents data from direct measurement. Meanwhile, by MAVG n, we will refer to data obtained from measurement data by replacing each point with the moving average of n points preceding and including it.

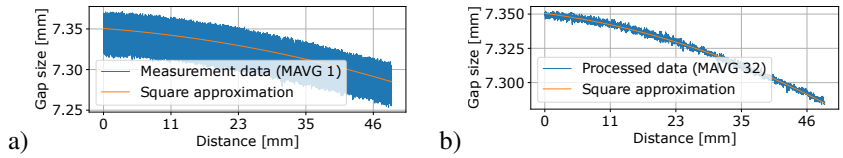


Figure 4. Data examples of the gap between the edge of the laser micrometer and the surface of the tested sample for: a) moving average size equal to 1, b) moving average size equal to 32

As the turning process is cyclical, it was assumed that the most suitable nature for the machine learning model would be data subjected to Fourier transformation. As a result of its operation, spectral data were obtained, revealing key frequencies characterizing the machining process. Figure 5 depicts the one-sided spectrum of the data presented in Figure 4 after subtracting the quadratic approximation from them. The machine learning process has been decided to be implemented

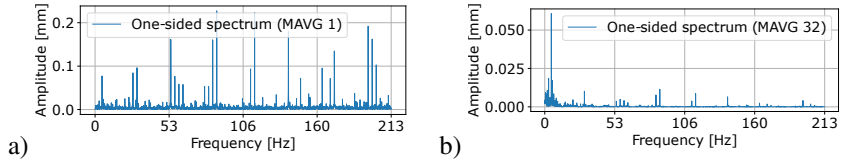


Figure 5. Examples of one-sided spectrum for the Fourier transform performed on 8192 elements for: a) moving average size equal to 1, b) moving average size equal to 32

based on the Support Vector Machine (SVM) method utilizing linear kernel classification (SVC). For the purpose of method verification, datasets $D(n, s, q)$ were created based on data from measurements conducted in a research setting. Parameter $n \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ indicates that moving average MAVG n was used in data preparation. On the other hand, $s \in \{256, 512, 1024, 2048, 4096, 8192\}$

represents the width of the data window for the Fourier transform used to generate the data. Meanwhile, $q \in \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\right\}$ denotes the q-th part of the lowest frequencies of the spectrum. The procedure for generating a dataset for a specified set of parameters is presented in Table 1.

Table 1. The procedure for generating a dataset $D(n, s, q)$

1. Set the values of parameters n, s, and q.
2. Calculate MAVG n for the original measurement data x_i, denoted as y_i.
3. For each y_i, calculate a quadratic approximation and subtract it to obtain z_i.
4. Select subsequences of length s from z_i and compute their Fourier transforms, creating sequences g_j.
5. Extract the lowest q frequencies from g_j, denoted as h_j.
6. Pairs (h_j, f_j), where f_j is the feed rate for a given sample, form the dataset $D(n, s, q)$.

Table 2. Model average errors in percentage for the Fourier transform performed for $s = 256$ (maximal errors) and $s = 8192$ (minimal errors)

FFT window size $s = 256$									
FFT part	Moving average size n								
q	1	2	4	8	16	32	64	128	256
1/2	1.229	1.277	1.322	1.334	1.163	1.012	1.083	11.258	20.993
1/4	1.246	1.283	1.322	1.336	1.162	1.012	1.083	11.277	21.000
1/8	1.373	1.371	1.366	1.340	1.160	1.014	1.082	11.292	21.003
1/16	1.337	1.323	1.306	1.289	1.157	1.016	1.081	11.319	21.024
FFT window size $s = 8192$									
FFT part	Moving average size n								
q	1	2	4	8	16	32	64	128	256
1/2	0.179	0.182	0.187	0.183	0.186	0.187	0.225	0.320	0.381
1/4	0.182	0.182	0.187	0.183	0.186	0.186	0.226	0.321	0.383
1/8	0.193	0.191	0.189	0.184	0.186	0.186	0.225	0.323	0.383
1/16	0.184	0.184	0.184	0.184	0.186	0.186	0.225	0.327	0.385

3. Results

The numerical tests were conducted using the scikit-learn library version 1.4.0. For each of the datasets $D(n, s, q)$, 1000 tests were conducted. Each test involving a random split of the dataset $D(n, s, q)$ into proportions of 80% for training the model and 20% for testing the model. The split was performed using the

`train_test_split` function, with the randomness seed being the number of the conducted test (ranging from 0 to 999). Table 2 presents the averaged percentage results of errors made by the model on the datasets $D(n, 256, q)$ and $D(n, 8192, q)$. After conducting the experiments, it was found that the SVM model exhibits the highest average error for the dataset $D(256, 256, \frac{1}{16})$. This error amounts to 21.024%, which corresponds to a model accuracy of 78.976%. Conversely, the SVM model achieves the lowest error for the dataset $D(1, 8192, \frac{1}{2})$, with an error of only 0.179%, resulting in an accuracy of 99.821%. The model errors averaged

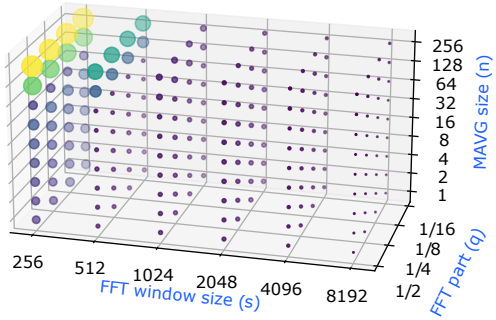


Figure 6. Model error for different parameters characterizing the measurement signal

over 1000 tests for all verified parameters are illustrated in Figure 6. The diameter and color of the points on the plot correspond to the magnitude of the average error for a given set of parameters. The presented chart indicates that the parameter q, which corresponds to the portion of the waveform subjected to analysis, does not have a significant impact on the average error made by the trained model. However, the other two parameters are highly significant. The smallest error is achieved for the largest FFT window size (parameter $s = 8192$) and MAVG 1 (parameter $n = 1$). In Figure 7, the change in the average error value and its standard deviation is presented based on the number of conducted tests. Cases for the minimum, median, and maximum standard deviation of the mean value are considered. In all plots, it is evident that the standard deviation tends to zero.

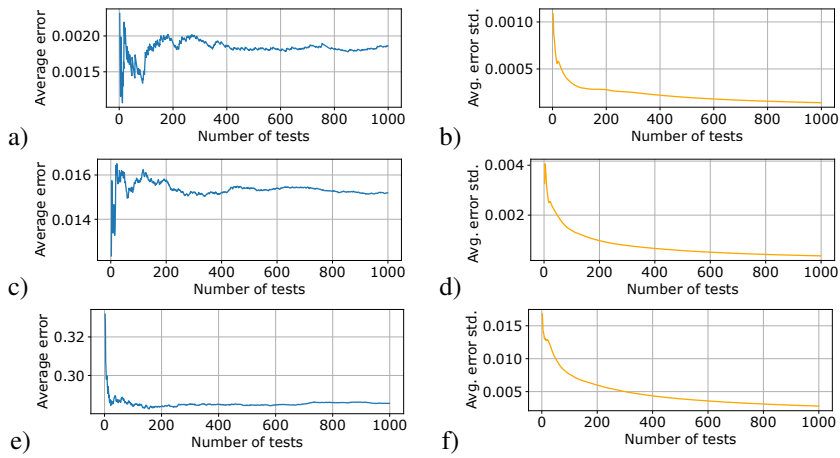


Figure 7. Average prediction error and standard deviation of the average prediction error for different sets tested $D(n, s, q)$: a), b) minimal deviation, c), d) median deviation, and e), f) maximum deviation

4. Conclusions

The article presents an innovative method for classifying feed sizes in the turning process using measurements obtained from an automatic laser micrometer. During the experiments, 216 sets of parameters were examined. Each set was used to prepare data for the Support Vector Machine (SVM) method with a linear classification kernel. The highest accuracy was achieved with the FFT window size ($s = 8192$), the widest spectrum width ($q = \frac{1}{2}$) and MAVG 1 ($n = 1$). The model error for this configuration amounted to 0.179%. This implies that the input data for the SVM model should be processed based on the parameters ($n = 1, s = 8192, q = \frac{1}{2}$), as they provide the highest effectiveness for the model.

References

[1] Zhang, B. and Shin, Y. A multimodal intelligent monitoring system for turning processes. *Journal of Manufacturing Processes*, 35:547–558, 2018. doi:10.1016/j.jmapro.2018.08.021.

[2] Cho, S., Asfour, S., Onar, A., and Kaundinya, N. Tool breakage detection using support vector machine learning in a milling process. *International Journal of Machine Tools and Manufacture*, 45(3):241–249, 2005. doi: 10.1016/j.ijmachtools.2004.08.016.

[3] Bhat, N., Dutta, S., Vashisth, T., Pal, S., Pal, S., and Sen, R. Tool condition monitoring by SVM classification of machined surface images in turning. *The International Journal of Advanced Manufacturing Technology*, 83:1487–1502, 2016. doi:10.1007/s00170-015-7441-3.

[4] Yang, S., Wang, S., Yi, L., Xue, H., Cao, Y., and Sun, S. A novel monitoring method for turning tool wear based on support vector machines. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 230(8):1359–1371, 2016. doi:10.1177/09544054166459.

[5] Jurkovic, Z., Cukor, G., Brezocnik, M., and Brajkovic, T. A comparison of machine learning methods for cutting parameters prediction in high speed turning process. *Journal of Intelligent Manufacturing*, 29:1683–1693, 2018. doi:10.1007/s10845-016-1206-1.

[6] Papandrea, P., Frigieri, E., Maia, P., Oliveira, L., and Paiva, A. Surface roughness diagnosis in hard turning using acoustic signals and support vector machine: A PCA-based approach. *Applied Acoustics*, 159:107102, 2020. doi: 10.1016/j.apacoust.2019.107102.

[7] Guleria, V., Kumar, V., and Singh, P. Classification of surface roughness during turning of forged EN8 steel using vibration signal processing and support vector machine. *Engineering Research Express*, 4(1):015029, 2022. doi: 10.1088/2631-8695/ac57fa.

[8] Chmielowiec, A. Algorithm for error-free determination of the variance of all contiguous subsequences and fixed-length contiguous subsequences for a sequence of industrial measurement data. *Computational Statistics*, 36(4):2813–2840, 2021. doi:10.1007/s00180-021-01096-1.

[9] Chmielowiec, A. and Klich, L. Application of python libraries for variance, normal distribution and Weibull distribution analysis in diagnosing and operating production systems. *Diagnostyka*, 22(4):89–105, 2021. doi: 10.29354/diag/144479.

Transition Into Solar Elevation Angle Domain for Photovoltaic Power Generation Forecasting

Paweł Parczyk[1][0009−0004−3287−9520], **Robert Burduk**[1][0000−0002−3506−6611]

[1]*Wrocław University of Science and Technology*
Department of Systems and Computer Networks
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
pawel.parczyk@pwr.edu.pl, robert.burduk@pwr.edu.pl

Abstract. *Nowadays, we observe an energy transition in which fossil fuel plants are replaced with renewable energy sources. It leads to new challenges. One of them is the inability to control energy production. Therefore, yield forecasting has become a pressing problem. However, previous work related to the issue of forecasting yields from photovoltaic installations was based mainly on the observation of past data. Therefore, the research did not focus on the phenomena that form the production. This work presents an alternative approach to searching for the source of a phenomenon instead of observing the effects. The article presents an analysis of the course of the sun and its impact on yields. The experiments that were performed showed the competitiveness of the proposed method compared to the Long Short-Term Memory.*

Keywords: *Renewable Energy Sources (RES), Photovoltaic (PV), Photovoltaic forecasting, Solar elevation, Solar movement, Yield forecasting*

1. Introduction

In modern society, electricity plays a very important role. It could be said that society is addicted to it. Nowadays, it is hard to imagine life without access to energy. Therefore, the blackout has become a major threat with the potential to cause significant consequences [1]. Unfortunately, fossil fuels on which the power system is currently based lead to environmental pollution. Moreover, their resources are limited. To counteract these effects, we have started an energy transition [2], which is based on Renewable Energy Sources (RES).

DOI:10.17388/WUT.2024.0002.MiNI

However, RES also has its own limitations. Production from RES strongly depends on many factors such as location, temperature, dust occurrence, weather, and cloud cover [3]. In other words, wind turbines need wind, and photovoltaics need the sun to produce energy.

Additionally, photovoltaic production is much higher on long summer days than on short and cloudy winter days [4]. Apart from seasonal, long-term changes, solar power plants are also exposed to sudden changes in sunlight. It happens when weather conditions change rapidly. This may cause short but significant fluctuations in power, which introduce instability to the grid. Therefore, energy forecasting may allow us to overcome this issue.

This work presents a method for predicting yields from photovoltaic installations, which is based on the analysis of the impact of the sun's height above the horizon on power generation.

2. Model assumptions

As the study shows [5], topics related to photovoltaic forecasting have become popular. Scientists are focused on methods related to RNN, such as LSTM [6] or more complex BiLSTM-CNN models [7]. However, the most common approach is to look at the dataset and use measured values to form the models as a black box. In other words, they do not consider phenomena that form the data.

In this section, the sun's movement and its effect on production are described.

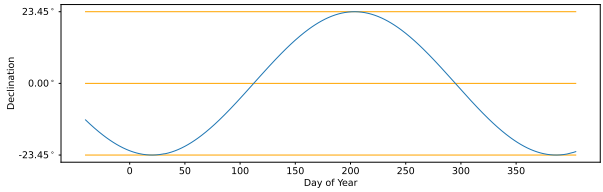


Figure 1. Declination angle during the year.

The sun's height above the horizon α depends on several factors: a constant one, a yearly and daily changing. The first is the latitude L of the location of the point on the Earth's surface. The second is the sun's declination angle δ, i.e., the planet's rotation axis angle from the plane of the normal to the disk along which the Earth orbits the sun. This value ranges from -23.45 to +23.45 degrees, and

changes occur annually. Moreover, it is the same for each location on the planet at the time. Figure 1 shows the behavior of the declination value for subsequent days of the year. We can observe these changes as the seasons of the year. The solar hour *HRA*, which changes daily. It is the third factor influencing the sun's altitude. The sun's height above the horizon for any moment of the year can be calculated from the equation 1.

$$sin(\alpha) = sin(L)sin(\delta) + cos(L)cos(\delta)cos(HRA) \tag{1}$$

Figure 2 portrays the annual course, omitting negative angles. There are four specific moments during the year. These are the summer solstice, winter solstice, and two equinoxes. Figure 3 compares the sun's altitude on those specific days.

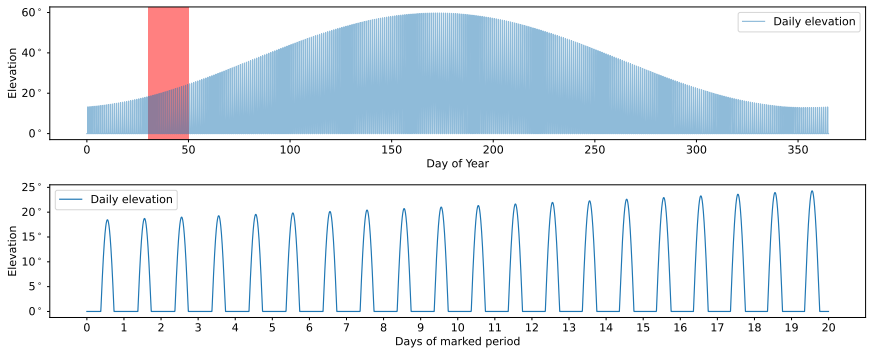


Figure 2. Elevation angle during the year (top) and a closer look for red marked rage (bottom) for a location with a latitude of 53.671^o.

The time-based approach (such as RNN-based models) is based on the relation between future and past data in the time domain. This relation can be described by equation $Y_{t+1} = M_1(time_t, \theta) + \epsilon_t$. Based on the presented phenomena, we can assume that the production is a function of the sun's position rather than time. Therefore, we propose transitioning from the time domain to the solar elevation angle domain. In that case the problem can be reformulated as $Y_{t+1} = M_2(elevation(time_t), \theta) + \epsilon_t \rightarrow Y_{t+1} = M_2(\alpha_t, \theta) + \epsilon_t$.

The principle of operation of the model is based on assigning subsequent production values Y_t to groups depending on the elevation angle at which they occurred.

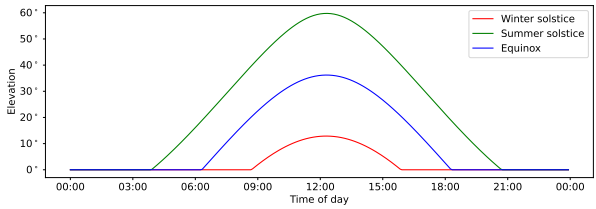


Figure 3. Comparison of Sun's elevation during most characteristic days: summer solstice, equinoxes, and winter solstice for a location with a latitude of 53.671^{o}.

First, during the model fitting, the sun's elevation angle is calculated for the selected period. The *window_size* parameter defines the length. Then, the angle values are quantized. This process assigns similar elevation angles to the individual corresponding groups. For example, assume that the maximum available sun elevation angle for the installation under analysis is 60 degrees and the minimum is 0 degrees. The number of bins is specified by *elevation_angle_bins*. When divided into six groups, each group will cover a range of 10 degrees (the first group will be from 0 to 10 degrees, the second from 10 to 20, etc.). Then, all values are aggregated into one value. A weighted average of all Y_t values belonging to each group is calculated. Those averages constitute the model θ.

Forecasting involves assigning production values to future points in time. This is done in a similar way to fitting. First, the sun's elevation angles in the processed period are calculated, and then values are assigned to these angles. This forecast becomes a stair-like forecast. Then, to eliminate this effect, the forecast is smoothed using linear interpolation. An example of a 6 hours ahead forecast is presented in Figure 4. The forecast was made for two consecutive days, one of which is sunny and the other is a day with temporary cloud cover. The upper chart presents a stair-like forecast, while the lower presents a smooth one. Moreover, it can be observed that for extremely good weather, the model underestimates the forecast, while for fine weather, it is much more precise.

3. Evaluation

Tests were performed to verify the effectiveness of the method proposed. Tests were carried out using the rolling window scheme with a constant window size.

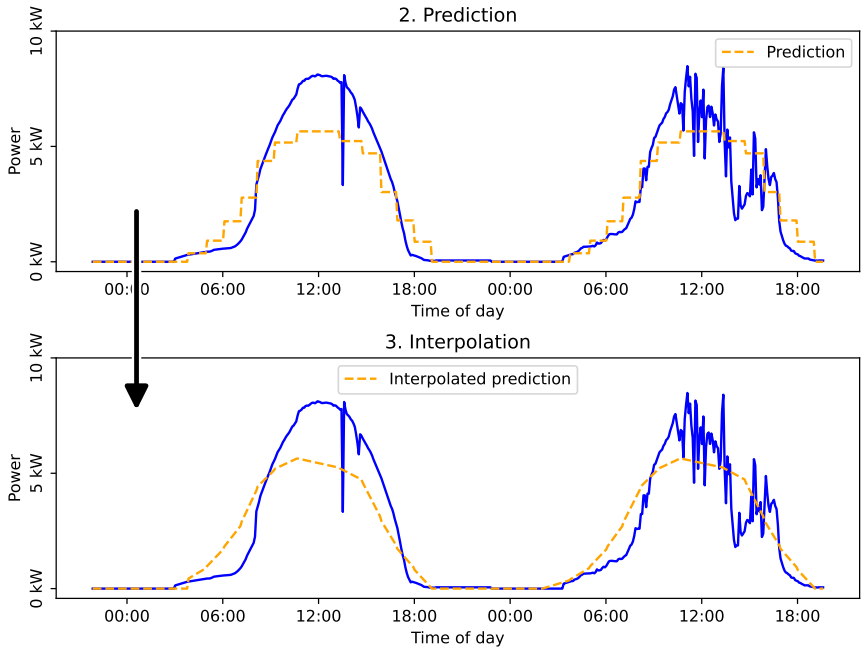


Figure 4. Sample 6 hours ahead forecast (the upper graph: generated from calculated weighted averages for each group of elevation angles) and forecast after applying linear interpolation (the lower graph: forecast after interpolation) for an exemplary period of two consecutive days containing a sunny day (from the left) and a day with transient cloudiness (from right).

The proposed method was compared to the state-of-the-art one-layer LSTM model. The comparison was performed on a high-resolution (5-minute sampling) dataset. Obtained from two different photovoltaic installations installed in Poland. The first installation is 9.9kWp, and the second is 6.12kWp. Both installations face south. The Dataset spans between 21st June 2020 (summer solstice) and 21st December 2023 (winter solstice) therefore, it contains 368350 1-dimensional samples (a power generated in kilowatts) from each installation.

The models were used to generate forecasts in three different time horizons. The first one is a short-term forecast for 6 hours ahead. The second horizon is

Table 1. List of considered proposed hyperparameters. In bold, selected values are marked.

Proposed model	
window_size	7 days, 14 days, 30 days, 45 days, 60 days, **75 days**, 90 days
elevation_angle_bins	10, 26, **40**, 50, 60

Table 2. List of considered reference model hyperparameters. In bold, selected are marked.

LSTM	
window_size	7 days, **14 days**, 30 days, 45 days, 60 days
inputs	2h, 4h, **6h**
max_epochs	25, 70, **100**, 130, 150
layer	**LSTM(32)**

longer and is 12 hours ahead. The third time horizon, which is 24 hours, can be classified as a medium-term forecast. The selection of horizons was based on their usefulness. Short-term and medium-term forecasts are useful for scheduling power plant operations.

A grid search method was used to find the optimal configuration of each model. During selection, models were tested on data covering one year. The considered combinations of hyperparameters are presented in Table 1 and Table 2. In each row in each table, the optimal parameter was marked with bold.

Table 3 presents the results of evaluating individual models on the database. The results were created by averaging the metrics values obtained by individual models during subsequent years (i.e., 2021, 2022, 2023). The evaluation was performed using the commonly used MAE, RMSE, NMAE, and R2 metrics.

Better values are bold in the table. It can be concluded that the proposed method copes better with short-term forecasting for 6 hours than the reference model. However, it is characterized by high variability of the obtained results, as indicated by the high standard deviation value. However, this relationship is not identical for both databases. Therefore, it can be said that the dispersion of results depends strongly on the entered data (with a larger installation, there is a greater dispersion of results) in the case of MEA and RMSE. The NMAE shows that the proposed method is better for 6 hours horizon and 12 hours horizon than LSTM.

Table 3. Yearly models' metrics that were achieved on processing various time horizons.

Model	Averaged annual prediction results					
	6h		12h		24h	
	MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
			D1			
LSTM	0.81 ± 0.032	1.47 ± 0.076	0.80 ± 0.065	1.46 ± 0.124	0.64 ± 0.038	1.33 ± 0.082
Proposed	**0.59 ± 0.139**	**1.30 ± 0.220**	**0.60 ± 0.141**	**1.31 ± 0.220**	**0.59 ± 0.141**	**1.31 ± 0.218**
			D2			
LSTM	0.44 ± 0.041	0.77 ± 0.066	0.45 ± 0.035	0.79 ± 0.051	0.35 ± 0.032	0.72 ± 0.064
Proposed	**0.33 ± 0.057**	**0.69 ± 0.080**	**0.34 ± 0.059**	**0.70 ± 0.086**	**0.34 ± 0.059**	**0.70 ± 0.087**

	6h		12h		24h	
	NMAE	R2	NMAE	R2	NMAE	R2
			D1			
LSTM	0.81 ± 0.100	**0.43 ± 0.034**	0.79 ± 0.058	**0.44 ± 0.037**	**0.65 ± 0.012**	0.48 ± 0.011
Proposed	**0.66 ± 0.054**	0.47 ± 0.078	**0.66 ± 0.057**	0.46 ± 0.084	0.66 ± 0.057	**0.46 ± 0.086**
			D2			
LSTM	0.82 ± 0.159	**0.42 ± 0.102**	0.76 ± 0.061	**0.47 ± 0.042**	**0.62 ± 0.030**	0.54 ± 0.030
Proposed	**0.64 ± 0.041**	0.53 ± 0.050	**0.64 ± 0.036**	0.52 ± 0.046	0.64 ± 0.036	**0.52 ± 0.047**

Additionally, it shows that the complexity of data in both datasets is almost the same. The obtained results for both datasets are close to each other regarding R2 and NMAE. In the case of 12-hour forecasts, the results of both models are closer to each other than for the 6-hour forecast, similar to the case of forecasts for the longest horizon. However, the proposed model remains the most effective for all forecast ranges discussed. The interesting fact, which needs deeper research is that the proposed method is better in terms of NMAE and worst in terms of R2 for the 6 and 12-hour horizon, while for the 24-hour horizon, the relation is reversed.

4. Conclusions

In this paper, an alternative approach to the issue of forecasting yields from photovoltaic installations is presented. The proposed method involves transitioning from the time domain to the sun elevation angle domain. Of course, the sun's elevation angle also depends on time and changes periodically. The proposed model was tested on a few years of data and, despite its simplicity, turned out to be competitive with the advanced reference LSTM model.

Moreover, the model is a good starting point for further research. The model is susceptible to development, and further steps will involve more extensive testing

and the proposal of alternatives for the weighted average aggregation function.

References

[1] Matthewman, S. and B., R. H. Blackouts: a sociology of electrical power failure. *Socialspace*, pages 1–25, 2014.

[2] Akpan, J. and Olanrewaju, O. Sustainable energy development: History and recent advances. *Energies*, 16(20), 2023. ISSN 1996-1073. doi:10.3390/en16207049. URL https://www.mdpi.com/1996-1073/16/20/7049.

[3] Bonkaney, A. L., Madougou, S., Adamou, R., et al. Impact of climatic parameters on the performance of solar photovoltaic (pv) module in niamey. *Smart Grid and Renewable Energy*, 8(12):379, 2017.

[4] Alhamer, E., Grigsby, A., and Mulford, R. The influence of seasonal cloud cover, ambient temperature and seasonal variations in daylight hours on the optimal pv panel tilt angle in the united states. *Energies*, 15(20), 2022. ISSN 1996-1073. doi:10.3390/en15207516. URL https://www.mdpi.com/1996-1073/15/20/7516.

[5] Iheanetu, K. J. Solar photovoltaic power forecasting: A review. *Sustainability*, 14(24), 2022. ISSN 2071-1050. doi:10.3390/su142417005. URL https://www.mdpi.com/2071-1050/14/24/17005.

[6] Gao, M., Li, J., Hong, F., and Long, D. Short-term forecasting of power production in a large-scale photovoltaic plant based on lstm. *Applied Sciences*, 9(15), 2019. ISSN 2076-3417. doi:10.3390/app9153192. URL https://www.mdpi.com/2076-3417/9/15/3192.

[7] Li, G., Guo, S., Li, X., and Cheng, C. Short-term forecasting approach based on bidirectional long short-term memory and convolutional neural network for regional photovoltaic power plants. *Sustainable Energy, Grids and Networks*, 34:101019, 2023. ISSN 2352-4677. doi:https://doi.org/10.1016/j.segan.2023.101019. URL https://www.sciencedirect.com/science/article/pii/S2352467723000279.

Ransomware Detection Using Machine Learning in the Linux Kernel

Adrian Brodzik[1][0009−0009−5980−0620],
Tomasz Malec-Kruszyński[1][0009−0009−9555−8560],
Wojciech Niewolski[1][0000−0001−6779−9038],
Mikołaj Tkaczyk[1][0009−0007−5462−6438],
Krzysztof Bocianiak[1],
Sok-Yen Loui[2]

[1]*Orange Polska, Aleje Jerozolimskie 160, 02-326 Warsaw, Poland*
[2]*Orange, 44 Avenue de la République, 92326 Châtillon, France*
firstname.lastname@orange.com

Abstract. *Linux-based cloud environments have become lucrative targets for ransomware attacks, employing various encryption schemes at unprecedented speeds. Addressing the urgency for real-time ransomware protection, we propose leveraging the extended Berkeley Packet Filter (eBPF) to collect system call information regarding active processes and infer about the data directly at the kernel level. In this study, we implement two Machine Learning (ML) models in eBPF - a decision tree and a multilayer perceptron. Benchmarking latency and accuracy against their user space counterparts, our findings underscore the efficacy of this approach.*
Keywords: *linux kernel, machine learning, ransomware detection*

1. Introduction

In recent times, there has been a notable surge in global cyberattacks, with ransomware emerging as a prominent threat. This malicious software encrypts user data and demands a ransom for its recovery. Modern variants are also capable of data theft and self-propagation. Globally, over 72% of all organizations claim to have been affected by ransomware in 2023 - this number is not expected to decrease [1]. In order to defend against such attacks, a low-level, efficient and

DOI:10.17388/WUT.2024.0002.MiNI

extremely low latency tool has to be used. The extended Berkeley Packet Filter (eBPF) fits those criteria perfectly, as it allows developers to run programs within the operating system kernel without having to modify the kernel source code.

The eBPF programs can be written in a constrained C-like non-Turing-complete programming language and compiled to eBPF bytecode using the Clang/LLVM toolchain and libbpf library. The bytecode is dynamically loaded into the kernel's virtual machine and checked in terms of execution safety by the eBPF verifier. This simplifies the development process, as well as ensures kernel stability after deployment. Although eBPF programs are stateless, special data structures, called BPF maps, can be created, allowing data storage and memory sharing. This enables communication between eBPF programs and user space applications. The fundamental purpose of eBPF is system observability. Different events can be captured using kernel probes, user probes, and other tracepoints. Apart from that, the Linux Security Module (LSM) framework introduces various security checks which can be hooked. These can be used to block specific user and process operations, allowing the creation of dynamic security policies. As such, combining eBPF with Machine Learning (ML) algorithms may offer enhanced detection capabilities. In this paper, we propose an innovative solution based on ML models embedded in eBPF programs for the purpose of real-time ransomware detection.

2. Related Work

Prior research has extensively explored the domain of ransomware detection, especially for Windows-based operating systems [2, 3]. The methods used range from static, dynamic, and hybrid analysis techniques, including ML models [4, 5]. Deceptive techniques like deploying Honeypots and Honeyfiles have also proved to be effective early detection mechanisms [6]. Currently, the use of ML systems is considered to be the most effective (especially against zero-day attacks). Different features can be utilized, such as those extracted from network traffic [7].

Many ML-based solutions are typically implemented in high-level programming languages and consequently reside in the user space. This placement introduces substantial latencies in the decision-making process. Given the speed of ransomware cryptographic algorithms, even seemingly inconsequential delays can lead to a higher volume of encrypted files. Optimization of the time between data acquisition and the final decision is critical. The use of ML in eBPF can be a significant step forward in this regard. Currently only a couple research papers

exist that touch on this subject; many of which use eBPF for process observability and ML inference in the user space [8]. The closest implementation involves the use of eBPF in an Intrusion Detection System (IDS), demonstrating enhanced performance of decision tree models in the kernel space compared to their user space versions [9]. This improvement is also observed for Random Forest, Support Vector Machine (SVM), and TwinSVM written in eBPF [10]. Both of these studies rely on network communication and focus on general intrusion detection. Our solution, on the other hand, explores the ransomware-specific threat model and utilizes features extracted from process activity.

The lack of research papers may serve both as a challenge and an opportunity for future research work in the rapidly expanding domain of ransomware attacks. Due to that fact, addressing the limitation of existing literature may unlock new insights and bring meaningful advancement within the field. Effectively optimizing decision-making through ML could mitigate the impact of future attacks, making them less detrimental and more challenging for potential adversaries to conduct.

3. Ransomware Detection in eBPF

More and more ransomware attacks are targeting Linux-based cloud environments. Therefore, it is essential to develop appropriate security measures to protect data from encryption and exfiltration. Instead of analyzing executable files, the focus is shifted on the behavioral aspects of ransomware. We assume that an attacker has compromised a developer account, can execute arbitrary commands, but does not have root access. The ransomware attack does not have to be a single executable binary file – it can be multiple commands (or a script) utilizing benign system administrative tools. Ransomware can be multi-threaded and multi-process, and can implement various techniques and encryption schemes.

3.1. Implementation

Our ransomware detection application protects a set of directory paths, e.g. containerized application data volumes. It is based on several key components. First, eBPF is used to monitor and count selected system calls and other events triggered by different processes. To overcome the overwhelming stream of information the data is filtered by the directory path it pertains to or by process ID which has accessed a protected path in the past. These events were selected based on prior ransomware analysis efforts and include: *file_permission*, *file_open*, *inode_create*,

inode_unlink, *inode_rmdir*, *inode_rename*, *getdents64*, *vfs_read*, *vfs_write*. More information can be found in the LSM[1] and Syscall[2] documentation. Additionally, Shannon entropy and Pearson's χ^2 goodness-of-fit metric are calculated for write operations. Secondly, the data is relayed from the kernel space to user space applications via eBPF data structure maps, such as ring buffers. There, the events can be processed using ML algorithms to distinguish benign and malicious activity. Finally, the process or entire user account can be blocked, access to a particular path can be revoked via traditional Linux capabilities or the Linux Security Module (LSM) framework. This method may introduce significant delays, which jeopardizes file integrity, given the speed of modern ransomware encryption schemes. Instead of sending event data and analyzing it in the user space, we attempt to implement the entire threat detection process directly in eBPF – reducing latency without losing accuracy. An overview of the architecture is shown in Figure 1.

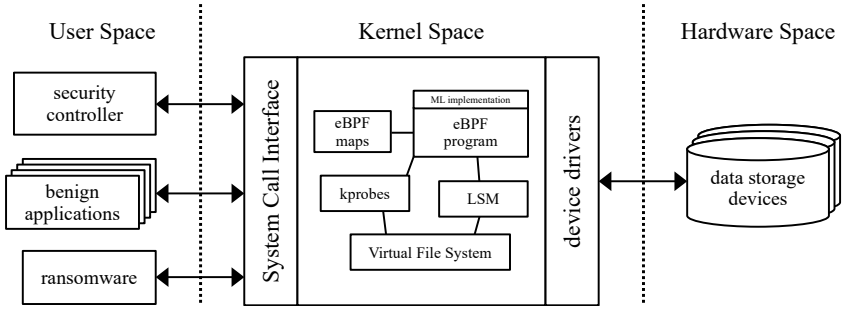


Figure 1. Real-time ransomware detection architecture overview.

For the purpose of this study, a decision tree and a multilayer perceptron were trained in a supervised matter to classify process events as either benign or malicious. The models were prepared using Scikit-Learn and PyTorch. Internally, a single decision tree structure is stored as several arrays, such as the feature used for splitting the node, the identifiers of the left and right child nodes, and the threshold value used to decide which child node to select next. The decision tree classifier was trained using default parameters with no depth or node count limitations. The final decision tree has 97 nodes. Similarly, PyTorch stores the parameters of each linear layer in their corresponding weight and bias tensors, which can be repre-

[1] `https://www.kernel.org/doc/html/v5.1/security/LSM.html`
[2] `https://www.man7.org/linux/man-pages/man2/syscalls.2.html`

sented as 1D and 2D arrays. The neural network was trained by minimizing binary cross-entropy loss using Adam optimizer with learning rate $\gamma = 10^{-4}$ for no more than 10000 epochs (until loss stopped decreasing). The final neural network has two linear layers ($12 \rightarrow 8$ and $8 \rightarrow 1$ features) with ReLU and sigmoid activation functions, totaling 113 parameters. All aforementioned models were retrained at least 20 times with different random number generator seeds to ensure consistency.

These models were chosen because of their simplicity and efficiency, while granting the ability to represent nonlinear data relations. Their functionality was replicated in C and eBPF, including small matrix multiplication and array-based tree traversal. Because eBPF is a restrictive programming environment, complex models like deep neural networks may be near impossible to implement. Decision trees and small neural networks, on the other hand, can be easily constructed and executed within the eBPF constraints, making them suitable for such use cases. Due to the limitations of eBPF, including the lack of floating-point operations, the models had to be adapted. All floating-point operations had to be converted to fixed-point arithmetic. Furthermore, the sigmoid function was not implemented, instead its input logits were used for thresholding. Model parameters were dynamically loaded using eBPF data structures or stored as hard-coded values in the read-only data section of the program.

3.2. Methodology

The aforementioned ransomware detection models were tested in a Proxmox virtual environment with two virtual machines. The first machine served as a single cluster node, running a database application that we wished to protect. The second machine acted as a benign traffic generator, sending HTTP requests to read, write, and delete random files of various sizes (1 KB - 1 GB), random names, and realistic file extensions (office documents, images, source code files). The contents were comprised of randomly generated Lorem Ipsum texts and sometimes compressed using LZMA. A dataset of 10 Linux ransomware samples was created and split into train and test sets, including Royal, Conti, Monti, HelloKitty, Kuiper, IceFire, RansomEXX, Buhti, BlackBasta, Hive, Cl0p, RedAlert. These samples were obtained from the MalwareBazaar database and represent several infamous Ransomware as a Service (RaaS) threat actors, which have started infiltrating Linux, Cloud, VMware ESXi, and similar environments, having previously targeted Windows-based systems. The chosen families have been analyzed in terms of historical significance. This way we achieve diversity in evasion, prop-

agation, enumeration, and encryption techniques. The malicious programs were run independently one by one with and without client traffic, as well as multiple samples simultaneously. After each experiment the test environment was rolled back to the previous snapshot. The objective was to measure the processing time and macro F_1 classification score across the different models in user space Python, C, and kernel space eBPF.

4. Experimental Results

For each test case, we recorded raw event data, process system call statistics, processing time, and predictions of our models. The final results have been aggregated in Table 1. Because the tests were conducted in a live environment, the sample size and dataset imbalance may vary (there are approximately twice as many malicious events). Overall, the eBPF implementations of the decision tree and multilayer perceptron yield a significant performance boost over their user space versions, without negatively affecting precision and recall. All models achieved macro $F_1 > 95\%$, accurately separating ransomware activity from this particular client network traffic simulator. As such, it would be beneficial to conduct more tests with varying traffic simulator algorithms and a less constrained environment, further masking ransomware activity, in the future.

Table 1. Decision tree and neural network experimental results.

			processing time (in nanoseconds)		
implementation	sample size	macro F_1	mean	standard deviation	median
decision tree					
Python	51601	0.997	167183	150271	150759
C	35400	0.998	818	1196	714
eBPF	36525	0.998	115	301	93
neural network					
Python	32165	0.953	94996	103528	83910
C	35410	0.965	1062	1304	754
eBPF	37910	0.974	220	488	180

Based on the average processing time, the eBPF implementation of the decision tree is 7.1x faster than the C version and 1453x faster than the Python version. Similarly, the eBPF implementation of the neural network is 4.8x faster than the C version and 431x faster than the Python version. The reason for such differences is because Python is an interpreted language, while C and eBPF are compiled. Even though the eBPF and C versions are almost identical, sending data from the kernel space, analyzing it in the user space, and sending back the classification result has a significant impact on the total processing time, delaying ransomware detection.

5. Conclusion

This paper investigates the integration of two ML algorithms, decision tree and multilayer perceptron, in eBPF with the objective of enhancing early ransomware detection in Linux environments. The results demonstrate the viability of this solution. In the presence of extremely fast encryption algorithms used by modern ransomware, this solution can be used to minimize latencies and enable real-time ransomware detection. Our ongoing research efforts focus on developing dedicated ML solutions tailored for the identification of ransomware activity. By implementing these models inside the Linux kernel we aim to achieve better detection results in terms of both effectiveness and time efficiency. Building upon the successful application of the decision tree classifier in the Linux kernel, our future work involves implementing other supervised and unsupervised algorithms in eBPF, such as Isolation Forests, Random Forests, and Support Vector Machines.

References

[1] IBM. Cost of a data breach report 2023. URL `https://www.ibm.com/reports/data-breach`.

[2] Hasan, R. A., AlSudani, A. M., and AlSudani, M. A. A survey on ransomware detection and mitigation techniques. *International Journal of Computer Science and Information Security*, 16(2):75–84, 2018.

[3] Urooj, U., Al-rimy, B. A. S., Zainal, A., Ghaleb, F. A., and Rassam, M. A. Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, 12(1):172, 2021.

[4] Alhawi, O. M., Baldwin, J., and Dehghantanha, A. Leveraging machine learning techniques for windows ransomware network traffic detection. *Cyber threat intelligence*, pages 93–106, 2018.

[5] Moussaileb, R., Cuppens, N., Lanet, J.-L., and Le Bouder, H. Ransomware network traffic analysis for pre-encryption alert. In *Foundations and Practice of Security: 12th International Symposium, FPS 2019, Toulouse, France, November 5–7, 2019*, pages 20–38. Springer, 2020.

[6] El-Kosairy, A. and Azer, M. A. Intrusion and ransomware detection system. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–7. IEEE, 2018.

[7] Almashhadani, A. O., Kaiiali, M., Sezer, S., and O'Kane, P. A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware. *IEEE access*, 7:47053–47067, 2019.

[8] Higuchi, K. and Kobayashi, R. Real-time defense system using ebpf for machine learning-based ransomware detection method. In *2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, pages 213–219. IEEE Computer Society, 2023.

[9] Bachl, M., Fabini, J., and Zseby, T. A flow-based ids using machine learning in ebpf. *arXiv preprint arXiv:2102.09980*, 2021.

[10] Anand, N., Saifulla, M., and Aakula, P. K. High-performance intrusion detection systemusing ebpf with machine learning algorithms. 2023.

Application of graph neural networks in particle track reconstruction

Wojciech Gomułka[1[0009−0003−2873−425X]],
Piotr A. Kowalski[2,3[0000−0002−4850−1758]],
Tomasz Szumlak[2[0000−0002−2562−7163]],
Tomasz Bołd[2[0000−0002−2432−411X]]

[1]*AGH University of Krakow*
Center of Excellence for Artifical Intelligence
al. Adama Mickiewicza 30, 30-059 Kraków
wgomulka@agh.edu.pl
[2]*Faculty of Physics and Applied Computer Science,*
AGH University of Krakow,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
{pkowal,szumlak,tbold}@agh.edu.pl
[3]*Systems Research Institute, Polish Academy of Sciences*
ul. Newelska 6, 01-447 Warsaw, Poland

Abstract. *This study describes the usage of graph neural networks (GNN) as one of potentially promising directions to address the problem of reconstructing the tracks of charged particles in high-energy physics. We point to previous attempts of such applications and use those approaches as entry point of our research. We present a proof-of-concept example of the usage of GNN in the problem, through the application of the basic architecture for the edge labeling task as a template. We also train and compare similar architectures using either SAGEConv, GCNConv or GatedGraphConv layers.*
Keywords: *high energy physics, artificial intelligence, particle track reconstruction, graph neural networks*

1. Introduction

High-energy physics (HEP) remains one of the most prominent examples of interdisciplinary research among physicists, computer and electronics scientists.

DOI:10.17388/WUT.2024.0002.MiNI

Every new generation of experiments pose a new challenges that still demand for novelty in software that helps in research. A particular use case is fast (near runtime) labeling of the data. It is already challenging and the data volumes are expected to grow even more in the future. Specifically, an incoming High-Lumi upgrade of Large Hadron Collider (LHC) which shall finish within a few years is hoped to increase the number of collisions by a factor of 5 - 7.5 [1].

In this paper, we focus on the topic of track reconstruction of charged particles, for which many different techniques have been used till now. They are taking into account physical reality of the process, like the helical movement of charged particles in a uniform magnetic field, with well-known algorithms, such as decision trees [1], Kalman filter [2], and the Hough transform [3]. Some of those approaches have a well-documented history of applications in tracking subjects in different domains and were also used in the field of particle physics. Hardware might be an essential limitation for application of these techniques, but recent interest in artificial intelligence combined with a growth of suitable computing powers might encourage more research in this direction. One of possibilities is the application of graph neural networks (GNN). For example, *X.Ju et al., 2020* [4] prepared GNN, which used differences between the coordinates of pairs of measurements left by particles in the detector to form GNN edge features and the measurement themselves as nodes. Track reconstruction was then treated as task of binary edge labeling on the pre-filtered set of hits. The concept of using GNNs for HEP tracking was also comprehensively described in *Duarte et al., 2022* [5]. We would like to leverage this idea to construct own architecture of GNN to address the problem of track reconstruction and present the initial research on the matter.

2. Idea of particle track reconstruction

High energy experiments, such as those at the LHC consist of many detector elements used to register signals of particle passage. After proper mapping, they give information about the coordinates of the signal; however, to restore tracks of the particles, there is a need to associate them with specific hits and then obtain properties of the particle. The task is trivial for single particle in the detector, all signals belong to that particle[2]. In the experimental reality, however, there is a significant number of particles that leave the signals in the detector at the same

[1] According to: `https://home.cern/resources/faqs/high-luminosity-lhc`.

[2] Assuming absence of the detector noise.

time.

Several approaches are used to perform track reconstruction, and the tracking algorithm might consist of several steps using different computational techniques. For example, ACTS (*A Common Tracking Software*) toolset, divides the track reconstruction process into steps such as seeding, track finding, track fitting and ambiguity resolution [3]. Seeding means finding a few initial hits believed to belong to the specific particle. This step has a major impact on the further reconstruction process. The toolset also provides virtual detectors such as ODD (*Open Data Detector*) and simulations of collisions inside those detectors. Example hits generated in the simulation of ODD are shown in Figure 1 and 2. The scatter plot of hits allows to implicitly visualize the detector geometry.

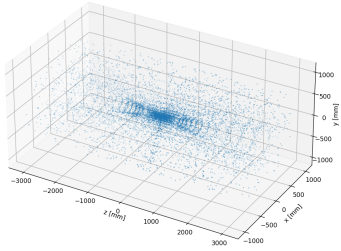


Figure 1. Example hits from simulation of the *OpenDataDetector* in the ACTS toolkit

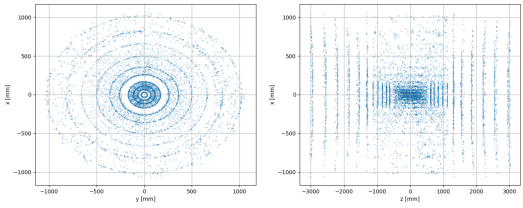


Figure 2. Example hits from simulation of the *OpenDataDetector* in the ACTS toolkit, projected on Y-X (left) and Z-X (right) planes

[3]Tracking explanation based on: `https://acts.readthedocs.io/en/latest/tracking.html`.

3. Graph neural networks

Graph neural networks [6] can address a wide range of problems, which can be translated to graphs. The rise in their popularity is relatively recent, yet already several different graph architectures have been designed and systematized in different surveys and review articles [6, 7]. The GNNs are significantly inspired by other architectures, among which graph convolutional networks (GCN), recurrent graph neural networks, or graph generative-adversarial networks (GGAN). Therefore, GNNs do not limit themselves to graph classification, but they can also be helpful at the node and edge level problems. The specific yet popular task is link prediction, which can be used in recommendation systems or social networking sites. Other popular applications of GNNs include, e.g. natural language processing, where GNNs can be helpful in modeling relations among text elements, by providing hierarchy [8], computer vision, where e.g. graph representation can support reasoning based on images or video frames [9], or chemistry, where graphs are well suited to represent the structure of the molecules.

In this paper, we opt for Graph Neural Networks, as every reconstructed particle track could be treated as a graph of the measurments forming it, as shown by [4]. The architecture of network proposed here and the training process used implementation of GNN from [10] as a template, upon which three similar architectures were build and compared.

4. Experiments

The three compared architectures differed by leveraged graph layers (SAGEConv, GCNConv or GatedGraphConv). GCNConv is an example of a spectral graph convolutional neural network [11]. SAGEConv (*sample and aggregate*) was introduced as a result of research related to the GCNConv. It is a graph neural network that addresses the problem of generating low-dimensional embeddings for node features in GNN [12]. Although convolutional GNNs seemed appropriate due to their ability to gather information from the neighborhood, recurrent neural networks (RNNs) were believed to be even more suitable, as consecutive hits of the specific track remain in the sequantial relation. Therefore, we use GatedGraphConv, which is a graph RNN inspired by the Gated Recurrent Unit (GRU) [13]. The general architecture of the models is presented in Figure 3.

The training was performed on the graph obtained from the ODD simulation of

a single collision event with about 3000 particles/labels. Therefore, the input graph consisted of approximately 200,000 nodes and 15,000 edges, from which half of the edges were labeled with positive class - edges between two consecutive hits belonging to the same track. The second half served as negative samples - edges generated between random hits. Although such an example is simplified when compared to real task of searching for the connections in the whole graph, yet it can still serve as a proof-of-concept for usage of GNNs in the tracking problem, and can be helpful for initial evaluation of the models. The training was carried out for 400 epochs in each case with a learning rate of 10^{-3} and ADAM chosen as optimizer. No early stopping was added. As a loss function, binary cross entropy with logits was selected. All the experiments were conducted with use of PyTorch Geometric framework. The final models were then evaluated on the few graphs built with different events generated from ODD. In case of SAGEConv and GCNConv, there were no significant differences from the testing/training metrics reached for the data from the original event. The results of data from one of such events is presented in Figure 5 in the form of confusion matrices. These data were not used for training of the models.

From Table 1. and Figure 4 we can argue that the model based on the SAGEConv or GCNConv layers outperformed the recurrent GatedGraphConv network in most of the metrics calculated. Model leveraging GatedGraphConv was converging faster, but last over 200 epochs had no significant impact on the loss.

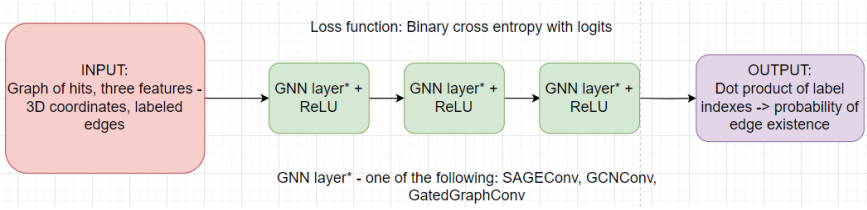


Figure 3. General architecture tested in the problem. For SAGEConv and GCNConv 256 channels were passed to the layers. For GatedGraphConv only one layer with 64 channels was used in the network to speed up the experiment

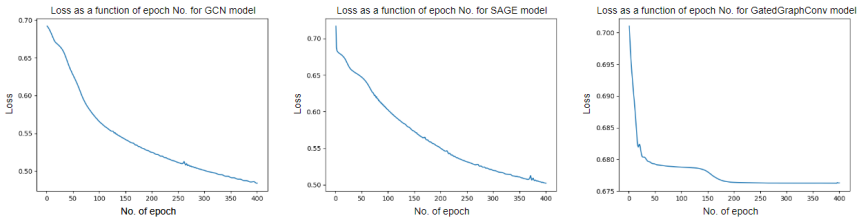


Figure 4. Training loss of the different models

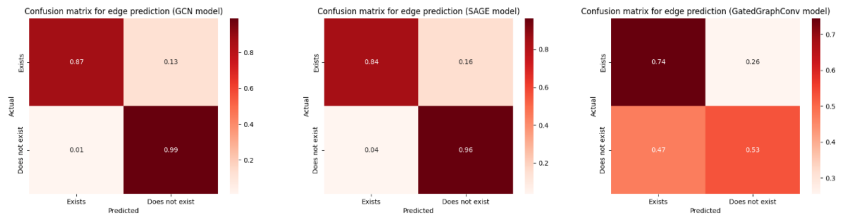


Figure 5. Confusion matrices for evaluation on example event

Table 1. Metrics for training of models using different layers

Metric	SAGEConv	GCNConv	GatedGraphConv
Sensitivity	0.976	0.993	0.402
Specificity	0.830	0.849	0.860
Precision	0.852	0.868	0.742
Negative predictive value	0.972	0.991	0.590
Accuracy	0.903	0.921	0.631

5. Conclusions

Usage of GatedGraphConv layer led to the least successful results among the compared approaches, despite the intuitive sequential nature of the hits in the track. Both GCNConv and SAGEConv allowed us to reach over 80% for all training metrics, with a slight edge of the first of the mentioned networks. It should be added, however, that the most important metric in measurements association is the efficiency (the ratio of the number good candidates/number of genuine tracks).

The results achieved in our experiments look promising in terms of GNN usage in the problem of particle tracking, despite the very initial phase of the research and simplified conditions of the task. Future studies will try to address the issue of track reconstruction in more realistic conditions, with taking advantage of filters inspired by physical and geometrical properties, which shall limit significantly the amount of hits considered for tracking. The training will be performed for bigger data volumes and different detectors with consideration of situations when there is no uniform magnetic field.

Acknowledgement

We gratefully acknowledge the funding support by program "Excellence initiative—research university" for the AGH University of Krakow as well as the ARTIQ project: UMO-2021/01/2/ST6/00004 and ARTIQ/0004/2021

References

[1] Davis, A., Dendek, A. M., De Cian, M., and Szumlak, T. PatLongLived-Tracking: a tracking algorithm for the reconstruction of the daughters of long-lived particles in LHCb. Technical report, 2017.

[2] Astier, P., Cardini, A., Cousins, R. D., Letessier-Selvon, A., Popov, B. A., and Vinogradova, T. Kalman filter track fits and track breakpoint analysis. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 450(1):138–154, 2000.

[3] Pozzobon, N., Montecassiano, F., and Zotto, P. A novel approach to hough transform for implementation in fast triggers. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 834:81–97, 2016.

[4] Ju, X., Farrell, S., Calafiura, P., Murnane, D., Prabhat, Gray, L., Klijnsma, T., Pedro, K., Cerati, G., Kowalkowski, J., Perdue, G., Spentzouris, P., Tran, N., Vlimant, J.-R., Zlokapa, A., Pata, J., Spiropulu, M., An, S., Aurisano, A., Hewes, V., Tsaris, A., Terao, K., and Usher, T. Graph neural networks for particle reconstruction in high energy physics detectors, 2020. doi:arXiv:2003.11603.

[5] Duarte, J. and Vlimant, J.-R. Graph neural networks for particle tracking and reconstruction. In *Artificial intelligence for high energy physics*, pages 387–436. World Scientific, 2022.

[6] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. doi:10.1109/TNNLS.2020.2978386.

[7] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[8] Wu, L., Chen, Y., Shen, K., Guo, X., Gao, H., Li, S., Pei, J., and Long, B. Graph neural networks for natural language processing: A survey, 2022.

[9] Chen, C., Wu, Y., Dai, Q., Zhou, H.-Y., Xu, M., Yang, S., Han, X., and Yu, Y. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective, 2022.

[10] Link prediction on heterogeneous graphs with PyG. URL `https://medium.com/@pytorch_geometric/link-prediction-on-heterogeneous-graphs-with-pyg`.

[11] Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks, 2017.

[12] Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs, 2018.

[13] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated graph sequence neural networks, 2017.

Comparison of tracking accuracy of HPE algorithms for sport speed climbing analysis

Adam Pieprzycki[1][0000−0001−7059−2118], **Adrian Ciochoń**[2][0009−0009−3470−961X], **Marcin Studniarz**[3][0009−0009−4582−9174], **Robert Rokowski**[4][0000−0003−1849−2587]

[1]*Department of Computer Science University of Applied Sciences in Tarnów*
Mickiewicza 8, 33-100 Tarnów, Poland
a_pieprzycki@anstar.edu.pl,
[2]*adrian.ciochon@o2.pl*
[3]*marstu99@outlook.com*
[4]*Department of Alpinism and Tourism, Faculty of Tourism and Recreation, University of Physical Education in Cracow*
rokofe@op.pl

Abstract. *The purpose of this study was to review the available HPE (Human Pose Estimation) algorithms. In addition, an analysis of the spatial-temporal parameters of athletes (climbers) was carried out based on video recording and the use of the geometrical algorithm Kanade-Lucas-Tomasi and two artificial intelligence- pre-trained - Deep Learning Convolutional Neural Network (CNN).*
Keywords: *HPE human pose estimation, sport speed climbing, Kanade-Lucas-Tomasi, K-T-L, Mediapipe, YOLO ver. 8, CNN.*

1. Introduction

Sport speed climbing is a sport that requires: perfectly synchronized movements, perfect technique, and lightning-fast decisions. Analyzing these movement activities with the help of advanced AI algorithms can help identify areas for improvement, optimize the climbing route, and increase efficiency. Such analysis can further develop time-climbing as an Olympic sport. One of the main challenges with the use of artificial intelligence is the analysis and application of algorithms

DOI:10.17388/WUT.2024.0002.MiNI

that can correctly analyze the climber's movement actions based on video recording. In addition, the algorithm used must be accurate and reliable.

Sport climbing in the Combined Olympic Format (lead, bouldering, and speed climbing) debuted at the 2020 Tokyo Olympics, delayed to 2021 due to the COVID-19 pandemic. At the 2024 Olympics in Paris, speed climbing will be a separate competition with two sets of medals, while the bouldering and lead competitions will be held as a new combined (duathlon) format event.

The climbing wall is constructed as a standard panel world record speed wall with a total height of 15.00 m with a continuous 5° overhang incline equipped with official speed timers and an official master hold provided by the IFSC (International Federation of Sport Climbing) for 20 identical handholds and 11 footholds [1].

2. Methods

In the article, the analyses considered were conducted on 15 videos - each with a different player, which were recorded with the same camera/lens from the same location. The study analysis was based on video recordings with the parameters: Camera type Sony A7 III / SIGMA Art 24mm F1.4 DG HSM, video resolution 1920×1080 (Full HD), video format mp4, camera parameters ISO 4000, F 3.4s 1/250, average illumination level - 375 lux under the climbing wall and 175 lux at the camera position (Sonel LXP-2 with LP-1 probe), distance of the camera to the wall - 10.0 m, height of camera positioning - 1.7 m, video frame rate - 100 fps, number of participants - 15 male (climbers of different sport levels).

3. Algorithms

Analysis of biomechanical parameters will be as good as an accurate determination of the displacement of a selected point on the body. For further analysis, the point chosen was the centre of the segment connecting the right and left hip plates - in the case of a climber, this will be the closest point identified with the centre of mass (gravity) BMC (body mass centre).

The primary task in the video analysis of the climber's movement was to determine position BMC x(t) and y(t), de-noising information (smoothening), and averaging such information.

Based on this, several different biomechanical parameters can be determined, such as [2]: total length of marker (BMC) trajectory (Euclidean distance) d_{tot}, total and two-axis component velocity, maximum (v_t^{max}) and minimum climbing velocity, accelerations (total a_t^{max}, and two-axis) obtained from video analysis: tangential acceleration, and centripetal acceleration (normal), the radius of curvature, convex hull $C(d_{tot})$(area) of the climbing run, entropy calculated as Global Index Entropy $GIE = \frac{log(2d_{tot} - log(C(d_{tot})))}{log(2)}$, geometric index of entropy $H = log(\frac{2d_{tot}}{C(d_{tot})})$, time to achieve maximum velocity ($t_{v^{max}}$) and position when achieving maximum speed, the potential (scaled height position/ location), kinetic, total energy of runni.ng and parameters of linear regression of achieved velocities $v_r = a \cdot t + b$.

All listed parameters except (knee, elbow, and hip) joint angles require precise determination of the athlete's BMC position. On the other hand, accurate extraction of key point detection human body skeleton affects the precision of the estimation of these joint angles.

The selected parameters cited in the paper are presented and discussed in more detail in Paper [2]. Morphological components and anthropometric characteristics [3] were not analyzed in this study, but may be relevant in the application of the information obtained during the analysis of the video recording.

Linear correction of perspective - the size of the recorded object carried out based on the size of the width of the structural panel of the wall (1.5 m). Since the recordings were carried out with the same parameters from the same place it was possible to correct the "pixel size" and camera information.

There are various solutions of (pre-trained - Deep Learning) neural networks that identify body key points of the human. Among the best-known are: Darkpose [4], AlphaPose [5], OpenPose [6] [7], FastPose (based on RestNet152), PosePipe [8], HyperPose, MediaPipe, YOLO (You Only Look Once). or other Human Pose Estimation (HPE) based e.g. on MPII Human Pose or COCO [9] or COCOv2 photo dataset. Access to such information allows for obtaining additional parameters useful for analyzing the climber's movement [10] in the biomechanical aspect or achieving joint mobility [11].

The work compares the total path of the selected BMC point using: non HPE geometric Kanade-Lucas-Tomasi (KLT) algorithm (with two markers/points) in Matlab 2020b environment [12] and Convolutional Neural Network (CNN) algorithms: MediaPipe (Python 3.10, PyCharm 2023.2.2) and YOLO ver.8 (Python 3.10.11, Visual Studio Code).

The skeleton of the MediaPipe algorithm consists of 33 key body points The points: These are distributed as follows: head: 11 points (nose, eyes, mouth), torso: 4 points (neck, shoulders, elbows, hips), upper limbs: 10 points (arms, hands), lower limbs: 8 points (thighs, knees, feet). The YOLOv8 human pose estimation model detects 17 key points: 5 key points for the spine, 4 key points for the left arm, 4 key points for the right arm, 2 key points for the left leg, and 2 key points for the right leg. Other algorithms: OpenPose 18 or 25 points, HyperPose 18 points, AlphaPose 17 and 26 points.

4. Results

Fifteen athletes were analyzed based on video recordings of their runs (Table 1). To compare the accuracy of analyzed algorithms, an r-Pearson linear correlation coefficient (Table 2) was calculated between the climber's total distance climbed (BMC) (Table 1, Figure 1) $d_{tot}^{algorithm}$ and the climbing time obtained from a certified digital measurement system installed in the wall infrastructure (launch platform and timer switch) minus the climber's reaction time ($t_r - t_{re}$).

The interpretation of which was based on indications [13]: ≤ 0.1 = trivial, 0.1 - 0.3 = small, 0.3 - 0.5 = moderate, 0.5 - 0.7 = large, 0.7 - 0.9 = very large, > 0.9 = almost perfect. Statistical significance of individual coefficients was assumed at $p < 0.05$. The obtained values of the correlation coefficient r-Pearson are statistically significant (Table 2) - test statistic: $t = r\sqrt{\frac{n-2}{1-r^2}}$, with an assumed Student's t-distribution (n-number of movies/athletes Table 1). The value of the correlation coefficient r-Pearson obtained with MediaPipe algorithm is much higher than that achieved with the K-L-T algorithm [2] (Table.2).

Although the analyzed accuracy affects the use of the various indicators mentioned earlier - direct application conclusions are difficult to apply. Even so, the coach can more accurately see the part of the run where the athlete loses too much time or the path of his center of gravity (BMC) is too "non-linear" (black box figure 1). With such an analysis, it is also possible to compare the passage of a given section of the route by different athletes, or by one athlete but with different methods of going. This can provide a starting point for modifying the trace of runs. Another application could be the estimation of maximum running speeds ($t_{v^{max}}$) that can translate to weight-bearing exercises in the gym.

Table 1. Parameters of equipment and experiments

No. movie/ athlete	**Running time** t_r **[s]**	**Reaction time** t_{re} **[s]**	**Total distance for algorithms**		
			K-T-L	**MediaPipe**	**YOLO ver.8**
			d_{tot}^{KTL} **[m]**	d_{tot}^{MP} **[m]**	d_{tot}^{YO} **[m]**
1	16.16	0.26	21.25	19.65	19.93
2	9.31	0.23	17.12	16.13	16.92
3	11.24	0.20	17.71	17.20	15.96
4	12.98	0.28	18.21	17.89	18.78
5	9.93	0.20	17.98	16.49	16.21
6	16.84	0.21	18.17	19.17	17.87
7	12.58	0.21	19.05	17.55	16.86
8	16.63	0.30	17.76	19.97	17.64
9	10.23	0.27	18.03	17.01	16.42
10	12.79	0.21	17.66	17.51	16.19
11	10.93	0.21	17.12	16.82	16.48
12	6.95	0.18	16.27	15.52	15.83
13	5.95	0.16	15.26	14.74	15.40
14	6.47	0.15	17.44	15.14	17.34
15	8.56	0.17	16,71	15.78	15.68

Table 2. Achieved values of the r-Pearson correlation coefficient

$t_r - t_{re}$ **[s]**	d_{tot}^{KTL} **[m]**	d_{tot}^{MP} **[m]**	d_{tot}^{YO} **[m]**
r**-Pearson**	0.7076	**0.9857**	0.6768
t - value	3.6093	21.0893	3.3145
p - value	0.0032	$1.9433\ 10^{-11}$	0.0056

5. Conclusion

Some aspects of the analysis: the athlete stands with his back to the camera (appropriate numbering of key points- hip positions are reversed), specific unchanging camera lens positioning correction of camera positioning and perspective of the recorded video (correction factor), automatic calculation of running time - detection of the start and end video frame, lack of visibility of limbs (obscured by the climber's body) during dynamic movements of the climber on the wall,

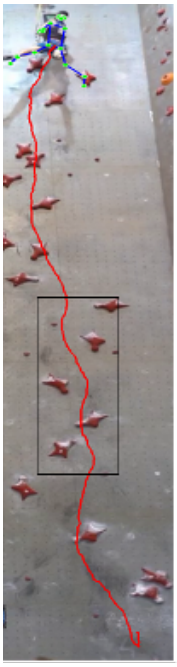


Figure 1. Visualization of the route taken by the climber

horizontal or vertical orientation of the analyzed video frames, switching between recognized people in the frame (Figure 1).

Improvements in accuracy can potentially be achieved by: marking the region of interest (ROI)- used in this study for the MediaPipe algorithm, film editing including: straightening, verticalizing, and horizontalizing the image, or using of tilt-shift lens. The authors conducted preliminary tests with the lens Samyang MF 3,5/24 T/S Sony E-Mount.

Other possibilities encountered in the literature review for changing the perspective of the study were the use of two cameras and 3D analysis, a rotating camera, or even the use of a drone moving in a horizontal plane. However, each of these techniques is a place to generate more errors. Using a hybrid measurement system located in the holds and footholds in combination with hands-point tracking algorithms can also be difficult. Potential errors in determining these points

could result from the lack of visibility of the hands hidden by the body contour. Other techniques would need to be used (e.g. Kalman filter).

In addition, there are errors during such analysis related to the algorithms themselves. Among such we can include missing frames or incorrect point detection.

One possible for future verification of the accuracy of estimation during speed-climbing can be the use UWB Ultra-Wideband (e.g. Murata Type2DK EVK: Trimension™ SR040 UWB). UWB is a broadband transmission technology that currently achieves an accuracy of 10 cm or Inertial Measurement Unit IMU devices (e. g. ICM-20948 Adafruit 4554). However, there is a lack of dedicated wireless equipment, and those that are available (Gyko Microgate, VmaxPro, Stryd, Myotest, Beast, PUSH) are more suitable for gym exercises. The accuracy of UWB and IMU was not analysed in this study.

An extension and potentially an improvement of the analysis extended to other algorithms could be the use of Docker hub software.

The very high linear r-Pearson correlation of a climber's running time and running distance (r=0.9876, α=0.05) for the MediaPipe algorithm may suggest the validity of its selection for analyzing spatial-temporal parameters in speed sport climbing.

It can be seen that the MediaPipe algorithm with the largest number of analysed human body points performed best with the analysis of a dynamically moving athlete specifically positioned with his back to the camera.

Supporting information

Funding: This work was supported by the University of Applied Sciences in Tarnow internal grant no. BAD-008/2021PWSZ/PRNR-s/0700-7/PN-U/2021.

Institutional Review Board Statement: The research protocol was approved by the local Bioethics Committee according to Resolution 6/0177/2019, and all research procedures were carried out in compliance with the Helsinki Declaration.

References

[1] IFSC, "Speed License Rules," 2022. [Online]. Available: https://cdn.ifsc-climbing.org/images/ifsc/Footer/Manufacturers/Speed_Licence_Rules_Walls.pdf. [Accessed 15 03 2022].

[2] A. Pieprzycki, T. Mazur, M. Krawczyk, D. Król, M. Witek and R. Rokowski, "Computer-aided methods for analysing run of speed climbers," Preprints MDPI, 2023.

[3] M. Krawczyk, M. Ozimek, R. Rokowski, M. Pociecha and P. Draga, "Anthropometric characteristics and anaerobic power of lower limbs and their relationships with race time in female speed climbers," in *SOCIETY. INTEGRATION. EDUCATION Proceedings of the International Scientific Conference*, Rezekne, Latvia, 2018.

[4] B. Xiao, H. Wu and Y. Wei, "Simple Baselines for Human Pose Estimation and Tracking," in *Computer Vision – ECCV 2018*, Cham,Switzerland,, 2018.

[5] H. S. Fang, S. Xie, Y. W. Tai and C. Lu, "RMPE: Regional Multi-person Pose Estimation," in *Proceedings of the International Conferenceon Computer Vision (ICCV)*, Venice, Italy, 2017.

[6] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei and Y. Sheik, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part AffinityFields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172-186, 2021.

[7] D. Pandurevic, P. Draga, A. Sutor and K. Hochradel, "Analysis of Competition and Training Videos of Speed Climbing Athletes Using Feature and Human Body Keypoint Detection Algorithms," *Sensors*, vol. 22, no. 6, 2022.

[8] R. J. Cotton, PosePipe: *Open-Source Human Pose Estimation Pipeline for Clinical Research*, arXiv, 2022.

[9] T.-Y. Lin, M. Maire, C. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick and P. Dollár, *Microsoft COCO: Common Objects in Context*, arXiv, 2015.

[10] R. Rokowski and T. Ręgwelski, Naukowe podstawy treningu we wspinaczce sportowej (Scientific basis for sport climbing training), Kraków: AWF Kraków, 2019.

[11] P. Draga, M. Ozimek, M. Krawczyk, R. Rokowski, M. Nowakowska, P. Ochwat, A. Jurczak and A. Stanula, "Importance and Diagnosis of Flexibility Preparation," *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, p. 2512, 07 04 2020.

[12] Matlab, "Face Detection and Tracking Using the KLT Algorithm - Matlab," 2021. [Online]. Available: https://www.mathworks.com/help/vision/ug/face-detection-and-tracking-using-the-klt-algorithm.html. [Accessed 19 12 2021].

[13] W. G. Hopkins, S. W. Marshall , A. M. Batterham and J. Hanin, "Progressive Statistics for Studies," *Sports Medicine and Exercise Science*, vol. 41, pp. 3-12, 2009.

Chapter 13

Young.AI

Track Chairs:

- mgr Stanisław Kaźmierczak - Warsaw University of Technology
- mgr Mikołaj Małkiński - Warsaw University of Technology
- dr Adam Żychowski - Warsaw University of Technology

Cartesian Genetic Programming Approach for Designing Convolutional Neural Network

Maciej Krzywda[1], **Szymon Łukasik**[12], **Amir H. Gandomi**[34]

[1]*Faculty of Physics and Applied Computer Science, AGH University of Krakow*
al. Mickiewicza 30, 30-059 Kraków, Poland
[2]*Systems Research Institute, Polish Academy of Sciences*
ul. Newelska 6, 01-447 Warsaw, Poland
[3] *Faculty of Engineering and IT, University of Technology Sydney 5 Broadway, Ultimo NSW 2007, Australia*
[4]*Department and Organization,*
University Research and Innovation Center (EKIK), Óbuda University
Bécsi út 96/B, Budapest, 1034, Hungary

Abstract. *The present study covers an approach to neural architecture search (NAS) using Cartesian genetic programming (CGP) for the design and optimization of Convolutional Neural Networks (CNNs). In designing artificial neural networks, one crucial aspect of the innovative approach is suggesting a novel neural architecture. Currently used architectures have mostly been developed manually by human experts, which is a time-consuming and error-prone process. In this work, we use pure Genetic Programming Approach to design CNNs, which employs only one genetic operation, i.e., mutation. In the course of preliminary experiments, our methodology yields promising results.*
Keywords: *Cartesian Genetic Programming, Convolutional Neural Networks, Neural Architecture Search*

1. Introduction

Neural Architecture Search (NAS)[1] has gained considerable traction as a fully automated approach in the design of Neural Networks Architecture. The method facilitates the generation of architectures that are not only comparable

DOI:10.17388/WUT.2024.0002.MiNI

but often superior in performance to those crafted manually. Essentially, NAS simplifies the traditional process where humans iteratively adjust neural networks through trial and error to identify successful configurations. Instead, it automates this process, unveiling more intricate structures. Comprising a spectrum of techniques and tools, NAS systematically evaluates numerous network architectures within a predefined search space. It employs a search strategy to choose the architecture that best fulfills the objectives of a specific problem, maximizing a fitness function. Despite its effectiveness, NAS poses significant computational and time-related challenges, exacerbated by the financial costs associated with utilizing graphics processing units (GPUs). Consequently, researchers and research groups are increasingly exploring alternative methods to optimize costs and identify the most efficient and effective neural network architecture tailored to address their specific research problems. This paper aims to investigate and present results for designing convolutional neural networks using Cartesian Genetic Programming (CGP).

2. Cartesian Genetic Programming for Convolutional Neural Network Design

Pure Cartesian Genetic Programming does not include the crossover operation, so the only genetic operation in CGP is a mutation. Using crossover is a hot topic of research to improve GCP, but is still widely investigated [2]. Several types of mutations can be applied in CGP, such as point mutation, gene mutation, and segment mutation. Point mutation involves randomly changing the value of a single gene in the genome, whereas gene mutation involves replacing an entire gene with a new one. Segment mutation involves replacing a genome segment with a new segment that may contain multiple genes[3, 4]. Here, we present the use of Cartesian Genetic Programming (CGP) for designing convolutional neural networks (CNNs). Our goal was to establish the architecture of the first neural network so that it could be compiled. Still, we did not define it according to existing solutions - we wanted the network design process to establish solution on its own. In our approach, the evolution process starts with a parent whose genotype is represented by a set of genes and information about which layers in the neural network are active. Then, two offsprings are created by mutating the parent's genotype, where mutations are performed with a specific mutation rate. Children resulting from the mutation process are judged to see whether they are better than

their parents, and if so, that child is the parent of the next generation. We have predefined ConvSets with usable layers, each of which is decoded. The randomly selected layer is replaced with another random layer from ConvSets. If a random number is generated that is the same as the original value (the layer selected to replace cannot be replaced by the same one), the function will generate a new random value. We then check that this change allows compilation and that all shapes between layers are correct. If so, the mutation process is complete. If the mutation results s not accepted, and we draw another set of genes to mutate. Then, each offspring is converted to a neural network, and their performance is compared with the parent. If any offspring achieves better performance than the parent, it is considered the new parent, and its genotype is used to generate the next generation of offspring. Otherwise, the parent remains unchanged. The comparison of mutations and performance is repeated for several generations. In each generation, the best genotype found is recorded and the value of the fitness function represents the best performance. If none of the offspring achieves better performance than the parent for several consecutive generations, a neutral mutation is applied to the parent and the evolution process continues. However, to keep the layers active, we need to check that the mutation will not affect the deactivation of the layers, which is essential for the correct operation of the neural network. If the mutation results in deactivating these layers, the mutation is not accepted, and we draw another set of genes to mutate. Then, each offspring is converted to a neural network, and their performance is compared with the parent. If any offspring achieves better performance than the parent, it is considered the new parent, and its genotype is used to generate the next generation of offspring. Otherwise, the parent remains unchanged. The comparison of mutations and performance is repeated for several generations. In each generation, the best genotype found is recorded and the value of the fitness function represents the best performance. If none of the offspring achieves better performance than the parent for several consecutive generations, a neutral mutation is applied to the parent and the evolution process continues. Our goal was and is still to explore potential spaces where we can discover the optimal network configuration, leveraging the evolution process along with the mutation process. This allows the neural network genotype to be optimized and aproaches us the best model for a given dataset. Fitness is defined as a measure of how well a particular neural network performs on a given task. In this example, the fitness function is the accuracy of the network in a validation dataset. The higher the accuracy, the higher the fitness of the network. During the evolution process, the goal is to maximize fitness by generating new candidate solutions (children) through

mutation and selecting the best ones for the next generation based on their fitness compared to their parents.

3. Experiments

3.1. Experimental Setting Computational Budget

In our research, we employed various activation functions, including Relu, Elu, Selu, Sigmoid, Softmax, Softplus, Softsign, Tanh, and Exponential. Additionally, we utilized the following pooling methods: GlobalAveragePooling2D, MaxPool2D, and AveragePooling2D. We applied a dropout rate of 0.2 and incorporated BatchNormalization with a momentum of 0.99 and epsilon of 0.001. The parameters of the GCP algorithm utilized in our experiments include rows set to 1, Cols equal to 30, Level-Back set to 10, and both Mutation rate and generation specified as values of the list [0.01, 0.05, 0.1] and [10, 25, 50] respectively.

$$Budget = P \times G \times E \times T \tag{1}$$

To compare our solution, we define the computational budget as follows (equation 1) where the population (P) was the count of chromosomes representing a single CNN. The evaluation epoch (E) is a count of the epoch to train the mutated CNN in each generation (G) iteration. Training epochs (T) track the best chromosome trained during the epoch after the last iterate of generation. We set the population as one. We define our budget, where in our experiments E (Evaluation Epochs) is 25. Evaluation Epochs are used in the evolution process when we try to decode CNN architectures that can be compiled and all shapes between layers are correct; then we train our architecture in 25 epochs, on a tiny part of the data (1000 records) (in a 70/30 ratio) and verify fitness. Then we compare the achieved fitness value with others achieved in a given generation, and the best individual (with the lowest fitness) is trained in 100 epochs (T).

In our approach, we use 10, 25, or 50 generations (G) and always 100 epochs (T) to train the best convolutional neural network. To avoid the overfitting effect, we implement an Early Stopping mechanism for monitoring loss function (Categorial Cross Entropy) with patience set as 10. Each of the best neural network architectures was trained 10 times and the final results were averaged and presented in Table 2. Data were shuffled each time in 80/20 proportion. Therefore, the possible computational budgets are as follows:

- For G=10: Budget = 1 * 10 * 25 * 100 = 25000 = **25K**
- For G=25: Budget = 1 * 25 * 25 * 100 = 62500 = **62.5K**
- For G=50: Budget = 1 * 50 * 25 * 100 = 125000 = **125K**

3.2. Experimental results

Table 1. Cartesian Genetic Programming parameters

Parameter	Rows	Cols	Level-Back	Mutation rate	Generation
Value	1	30	10	[0.01, 0.05, 0.1]	[10,25,50]

To validate the robustness of the results, an experiment was conducted in which the parameters listed in Table 1 were used, and the experiment was repeated ten times. The results of this experiment as CNNs stacked with layer recognition are presented in (Table 1). In the table presented (Table 2) we demonstrate our result compared to CNN-based selected baseline solutions using classification images from the MNIST, and Fashion-MNIST datasets depending on the defined budget of the proposal and depends on mutation rate (MR). We also present all the results obtained, depending on the defined budget and mutation ratio. We can see that the medium budget **62.5K** yields the best results, while for a budget of 125K, the algorithm performs the worst. The number of generations determines the budget size; therefore, you can conclude that the increase in generations influences the degradation of the results. Based on the experimental settings and results, we can define the same budget using a larger population of neural networks instead of increasing generations. But at this stage of research, it is only a reckless hypotesis, which needs more evidence including more run experiments with a different range of parameters, including a smoother range of Mutation rates and bigger population, not only one.

4. Conclusions

In the last few years, studies on designing Artificial Neural Networks (ANNs) have become an active research field, mainly due to the advanced cost training and prototyping of underlying deep learning architectures. Our approach, distinguished from more complex solutions [9, 10, 11], focuses on simplicity. We

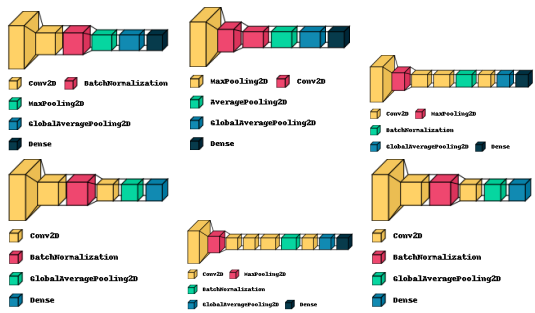


Figure 1. Visual representation of a stack of layers in Designed Convolutional Neural Network with CGP

employ basic CNN layers, disregard data dimensionality post flattening, and incorporate residual connections. Additionally, our method seamlessly integrates state-of-the-art models as input architecture, facilitating the analysis of changes in future generations.The proposed study focuses on developing the first stage of a method for the design and optimization of artificial neural networks based on Cartesian Genetic Programming. The focus is on developing an efficient and effective approach to designing and optimizing ANNs.

Acknowledgment

This work was partially supported by the program 'Excellence Initiative - Research University' for the AGH University of Krakow and by a Grant for Statutory Activity from the Faculty of Physics and Applied Computer Science of the AGH University of Krakow. We gratefully acknowledge the Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2022/015677"

References

[1] Elsken, T., Metzen, J. H., and Hutter, F. Neural architecture search: A survey. *ArXiv*, abs/1808.05377, 2018.

Table 2. Accuracy in % achieved by our models

Models	**MNIST**	**FashionMNIST**
Our CNN (**25K**,MR=**0.1**)	92.51 ∓0.351	86.87 ∓0.201
Our CNN (**62.5K**,MR=**0.1**)	97.28∓0.842	72.82∓0.435
Our CNN (**125K**,MR=**0.1**)	55.29∓1.113	44.82∓1.384
Our CNN (**25K**,MR=**0.01**)	97.15∓0.373	83.27∓0.883
Our CNN (**62.5K**,MR=**0.01**)	97.75∓0.220	70.95∓0.538
Our CNN (**125K**,MR=**0.01**)	54.43∓0.984	86.44 0.271
Our CNN (**25K**,MR=**0.05**)	97.92∓0.119	83.73∓0.339
Our CNN (**62.5K**,MR=**0.05**)	92.25∓0.237	72.10∓0.471
Our CNN (**125K**,MR=**0.05**)	91.82∓0.325	84.06∓0.792
L.Hertel et al. (2017)[5]	99.68	N/A
L.Wan et al. (2013) [6]	99.79	N/A
K.Meshkini et al. (2020) [7]	N/A	93.43
M.Kayed et al. (2020) [8]	N/A	98.80

[2] Husa, J. and Kalkreuth, R. A comparative study on crossover in cartesian genetic programming. In M. Castelli, L. Sekanina, M. Zhang, S. Cagnoni, and P. García-Sánchez, editors, *Genetic Programming*, pages 203–219. Springer International Publishing, Cham, 2018. ISBN 978-3-319-77553-1.

[3] Miller, J. F. Cartesian genetic programming: its status and future. *Genetic Programming and Evolvable Machines*, 21:129–168, 2019.

[4] Miller, J. F. *Cartesian Genetic Programming*, pages 17–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-17310-3. doi:10.1007/978-3-642-17310-3_2.

[5] Hertel, L., Barth, E., Käster, T., and Martinetz, T. Deep convolutional neural networks as generic feature extractors. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–4. 2015. doi:10.1109/IJCNN.2015.7280683.

[6] Wan, L., Zeiler, M. D., Zhang, S., LeCun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*. 2013.

[7] Kayed, M., Anter, A., and Mohamed, H. Classification of garments from fashion mnist dataset using cnn lenet-5 architecture. In *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pages 238–243. 2020. doi:10.1109/ITCE48509.2020.9047776.

[8] Meshkini, K., Platos, J., and Ghassemain, H. An analysis of convolutional neural network for fashion images classification (fashion-mnist). In S. Kovalev, V. Tarassov, V. Snasel, and A. Sukhanov, editors, *Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19)*, pages 85–95. Springer International Publishing, Cham, 2020. ISBN 978-3-030-50097-9.

[9] Torabi, A. R., Sharifi, A., and Teshnehlab, M. Using cartesian genetic programming approach with new crossover technique to design convolutional neural networks. *Neural Processing Letters*, 55:5451–5471, 2022.

[10] Suganuma, M., Kobayashi, M., Shirakawa, S., and Nagao, T. Evolution of deep convolutional neural networks using cartesian genetic programming. *Evolutionary Computation*, 28:141–163, 2020.

[11] Banerjee, S. and Mitra, S. Evolving optimal convolutional neural networks. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2677–2683, 2020.

Classifier Ensemble Dynamic Model with Conflict Analysis

Małgorzata Przybyła-Kasperek[1][0000−0003−0616−9694]
Jakub Sacewicz[1][0009−0003−1963−7079]

[1]*University of Silesia in Katowice*
Institute of Computer Science,
Będzińska 39, 41-200 Sosnowiec, Poland
malgorzata.przybyla-kasperek@us.edu.pl, jakub.sacewicz@us.edu.pl

Abstract. *This paper proposes a dynamic model for an ensemble of classifiers that uses data provided by independent sources. The data considered – dispersed data – is collected in several independent decision tables, in which both objects and attributes dispersion can occur. An ensemble of classifiers such as AdaBoost, Random Forest, or Random Subspace Method are built for each decision table separately. The compatibility of ensembles predictions for test objects is analyzed and coalitions of ensembles are formed in the proposed model. The final decision is made by the strongest coalition. The paper presents a model proposal and basic definitions, as well as some preliminary experimental results.*
Keywords: *Dispersed data, Conflict analysis, AdaBoost, Random Forests, Random Subspace Method, Ensemble of classifiers*

1. Introduction

Real-world data collection is often decentralized and dispersed, challenging machine learning in practical sectors like healthcare and banking, where recognizing the compatibility of decisions made from isolated data sets and the cooperation of classifiers is important. This study introduces Pawlak's conflict model to enhance conflict understanding in those scenarios. By dispersed data we mean a set of local decision tables autonomously gathered without unification of data, impossible to aggregate.

DOI:10.17388/WUT.2024.0002.MiNI

The concept of distributed data is considered in the literature as part of distributed learning [1], within which we consider federated learning [2]. In classical distributed learning approaches, the data is originally collected in a single data set and dispersed to improve the quality of classification. Such approaches include models: AdaBoost [3] – sensitizing base classifiers to particularly difficult cases; Random Forest [4] – ensembles of decision trees are built based on diversified local data; Random Subspace Method – diverse sets of conditional attributes in training data sets.

The analysis of conflicts and negotiation have evolved beyond traditional boundaries, thanks to the influence of game theory [5, 6]. The intersection of conflicts and Artificial Intelligence (AI) initially focused on decision support systems to assist individuals in better understanding conflicts, identifying conflicting issues, and exploring potential coalitions [7]. Significant contributor to the field of conflict analysis is the Pawlak conflict model, which is based on rough set theory [8, 9]. This model offers a foundational perspective on conflicts and has stimulated the development of various extensions and approaches. Notable researchers have specifically explored adaptations of the Pawlak model for multi-agent systems [10, 11]. Another noteworthy development is the three-way decision theory, which aligns with the tripartite agent values set (against, neutrality, and favorable) in the Pawlak model [12, 13]. This three-fold division has inspired research that examines conflicts through the trisection of agents, issues, and agent pairs, utilizing diverse evaluation functions [14, 15, 16, 17].

It should be noted that the recognition of compatibility between base classifiers and the use of coalitions and cooperation of such classifiers is very rare in the literature. There are few cases of such applications. In [18, 19] Pawlak's model a $k-$nearest neighbor classifier is used to detect alliances among base classifiers. In [20, 21] Pawlak's model with trees constructed through the bagging approach was investigated. Both mentioned approaches illustrated a positive impact on the classification's quality. This encourages us to continue exploring further possibilities of using a cooperative model of classifier ensemble. In this paper, we propose a model in which an ensemble of classifiers generated for each local table, generates prediction as a probability vector of an object belonging to one of the decision classes. Next, using vectors, coalitions of ensembles making similar decisions are generated with the use of Pawlak's conflict analysis model. Various approaches to determining the final decision were analyzed. In this paper, the strongest coalition makes the final decision. The novelty of this paper is the use of a combination of AdaBoost, Random Forest and Random Subspace Method with conflict analysis

and using coalitions of base models.

The rest of the paper is organized as follows. In Section 2, we present the details of the model and some preliminary experimental results. The last section is a conclusion and future plans.

2. Methods and models

In this paper, we assume that dispersed data is available, that is a set of local decision tables. More formally, there are local decision tables represented as $D_i = (U_i, A_i, d)$ for $i \in 1, \ldots, n$, where U_i represents the universe, a set of objects; A_i is a set of conditional attributes, describing features of the objects; and d denotes the decision attribute, which represents labels. Based on each table, the ensemble of classifiers is built. Out of any ensemble methods, this initial study applies three selected approaches. AdaBoost is an ensemble learning method that belongs to the boosting family. It employs weak classifiers, the most commonly used are decision trees with a depth of one (stumps). These weak classifiers are sequentially trained, with more emphasis on misclassified instances in each subsequent iteration. Random Forest falls under the bagging (Bootstrap Aggregating) family. It builds multiple decision trees using a random subset of the features and randomly selecting subsets of the training data with replacement. This ensures that each tree sees a slightly different version of the data set. The Random Subspace Method is a variation of the ensemble method that is based on the concept of feature subspaces. In this study, as a base classifier, we build a full tree based on a randomly selected subset of features and a full set of data instances. In all of the above approaches, the twenty base classifiers that make up the ensemble were used. The dynamic nature of the proposed system is revealed when the test object is classified. Each time we classify an object, conflict analysis and generating coalition formulate a different arrangement of classifiers. Then each ensemble generates a prediction vector that determines the probability of the object belonging to a given decision class. So each ensemble i based on one local table generates a prediction vector $[\mu_{i,1}(\hat{x}), \ldots, \mu_{i,c}(\hat{x})]$ with dimension c equal to the number of decision classes for the test object $\hat{x}$. The value $\mu_{i,j}(\hat{x})$ is the normalized number of votes cast for decision class j by the base classifier in the ensemble. These prediction vectors are then used in a conflict analysis model to recognize coalitions of ensembles. In the Pawlak model of conflict analysis, information about the conflict situation is stored in an information system $S = (LM, DecVal)$, where LM is a set of local models,

$DecVal = V^d$ is a set of decision attribute values. Function $dv : LM \to \{-1, 0, 1\}$ for each $dv \in DecVal$ is defined

$$dv(i) = \begin{cases} 1 & \text{if the coordinate } \mu_{i,v}(\hat{x}) \text{ for decision } v \in V^d \text{ in the prediction vector} \\ & \text{of ensemble } i \text{ has the maximum value of coordinates in the vector.} \\ 0 & \text{if the coordinate } \mu_{i,v}(\hat{x}) \text{ for decision } v \in V^d \text{ in the prediction vector} \\ & \text{of ensemble } i \text{ is the second highest of all coordinates in this vector.} \\ -1 & \text{in other cases} \end{cases}$$

In this manner, the ensembles express their opinions regarding the decision on the test object. In the subsequent step of the Pawlak conflict analysis model, a conflict function is utilized. The conflict function, denoted as $\rho : LM \times LM \to [0, 1]$, is defined as follows:

$$\rho(i, j) = \frac{card\{dv \in DecVal : dv(i) \neq dv(j)\}}{card\{DecVal\}},$$

where $card\{DecVal\}$ is the cardinality of the set of decision attribute values.

A set $X \subseteq LM$ is considered a coalition of ensembles if, for every $i, j \in X$, the conflict function satisfies $\rho(i, j) < 0.5$. This implies that the coalition includes ensembles whose opinions are consistent in more than half of the decision classes. Coalitions may not be disjoint sets, allowing an ensemble to belong to multiple coalitions simultaneously. In this study, only the most numerous coalition makes decisions. Then, each selected ensemble casts one vote for the decisions with its maximum support. Finally, the decision that receives the maximum number of votes is made.

The proposed system was tested on two data sets from the UC Irvine Machine Learning Repository [22, 23]. Lymphography, a medical imaging technique for visualizing the lymphatic system, essential for immune system functions, is characterized by four decision classes and 18 features. The training set encompasses 104 objects, with 44 objects in the test set. The Primary Tumor data set is tailored for medical diagnostics research, specifically in tumor classification and characterization. With 22 decision classes and 17 features, the training set includes 237 objects, while the test set comprises 102. All these data sets were initially available as one data set in the repository. However, they were dispersed into 3, 5 or 7 local tables, each containing a random subset of attributes, introducing diversity with some common attributes shared between tables. To assess the quality of the classification, the following measures were used: F1 weighted measure, balanced

Table 1. Results of F1 weighted measure (F1 (Weig.)), balanced accuracy (BAcc), classification accuracy (Acc), minimum accuracy (Acc min), maximum acccuracy (Acc max) and standard deviation (SD) for the dispersed data sets. LT means local tables; EnAdaBoostCoal, EnRandomForestCoal, EnRandomSubspaceCoal means Ensemble AdaBoost or RandomForest or RandomSubspace with coalitions; MajVote means Majority vote

Data set no. of tables	Model	F1 (Weig.)	BAacc	Acc	Acc min	Acc max	SD
Lympho-graphy 3LT	EnAdaBoostCoal	0.265	0.295	0.402	0.386	0.43	0.019
	EnRandomForestCoal	0.282	0.325	0.443	0.365	0.445	0.029
	EnRandomSubspaceCoal	**0.289**	**0.363**	0.452	0.409	**0.477**	0.017
	DecisionTreeMajVote	0.268	0.300	0.409	0.409	0.409	0.0
	RandomForestMajVote	**0.289**	0.333	**0.455**	**0.455**	0.455	0.0
Lympho-graphy 5LT	EnAdaBoostCoal	0.278	0.317	0.432	0.409	**0.455**	0.019
	EnRandomForestCoal	0.282	0.296	0.407	0.387	0.432	0.017
	EnRandomSubspaceCoal	**0.284**	**0.333**	**0.455**	**0.455**	**0.455**	0.0
	DecisionTreeMajVote	0.253	0.283	0.386	0.386	0.386	0.0
	RandomForestMajVote	0.266	0.297	0.405	0.387	0.409	0.012
Lympho-graphy 7LT	EnAdaBoostCoal	0.230	0.243	0.332	0.318	0.364	0.016
	EnRandomForestCoal	0.288	0.316	0.432	0.409	0.455	0.019
	EnRandomSubspaceCoal	0.297	**0.336**	**0.459**	**0.455**	**0.477**	0.010
	DecisionTreeMajVote	**0.320**	0.331	0.455	**0.455**	0.455	0.0
	RandomForestMajVote	0.274	0.310	0.423	0.409	0.432	0.012
Primary Tumor 3LT	EnAdaBoostCoal	**0.131**	0.063	0.165	0.059	0.127	0.024
	EnRandomForestCoal	0.123	0.051	0.141	0.118	0.177	0.022
	EnRandomSubspaceCoal	0.103	0.038	**0.173**	0.118	**0.216**	0.033
	DecisionTreeMajVote	0.069	0.018	0.088	0.088	0.088	0.0
	RandomForestMajVote	0.130	**0.064**	0.164	**0.137**	0.186	0.017
Primary Tumor 5LT	EnAdaBoostCoal	0.114	0.045	0.210	0.206	0.216	0.005
	EnRandomForestCoal	0.087	0.034	0.112	0.059	0.148	0.031
	EnRandomSubspaceCoal	**0.122**	**0.049**	**0.244**	**0.216**	**0.284**	0.021
	DecisionTreeMajVote	0.076	0.024	0.118	0.118	0.118	0.0
	RandomForestMajVote	0.100	0.035	0.125	0.108	0.137	0.011
Primary Tumor 7LT	EnAdaBoostCoal	0.135	0.059	0.294	0.294	0.294	0.0
	EnRandomForestCoal	0.172	0.070	0.303	0.265	**0.324**	0.018
	EnRandomSubspaceCoal	0.130	0.057	0.276	0.235	0.294	0.021
	DecisionTreeMajVote	**0.204**	**0.092**	**0.314**	**0.314**	0.314	0.0
	RandomForestMajVote	0.177	0.072	0.310	0.304	0.323	0.007

accuracy and classification accuracy. The F1 weighted score incorporates the individual F1 score for each class, but when computing the average, it assigns a weight based on the count of true instances for each class. This measure considers the issue of class imbalance. Since the proposed methods are non-deterministic due to randomness in selecting attributes and objects for training sets based on which trees are built, all experiments were performed 10 times. In Table 1, the average

values from these 10 performances are given. The proposed dynamic model with coalitions was compared with the approach, in which coalitions were not recognized – base model was built based on each local decision table separately and the final decision was made by majority voting. Two different approaches were used to build the base model: a single decision tree (the Gini index was used as the splitting criterion and no pre-pruning was applied) and a random forest with twenty decision trees. The results obtained are also given in Table 1.

The best results in terms of F1-weighted measure, balanced accuracy and accuracy are bolded in the table. As can be seen, in the majority of cases using conflict analysis and forming coalitions of ensemble improved the quality of classification. This is particularly evident in the Primary Tumor 5LT test case where classification accuracy was two times higher. Of course, these are preliminary experiments that need to be expanded and further comparisons should be made. However, these preliminary results show improvement in classification quality.

3. Conclusions

The paper proposes a classification model for dispersed data in which an ensemble of classifiers is built for each local table. The novelty of the model is the use of conflict analysis and recognition of the consistency of decisions (prediction vectors) generated by the ensembles. An approach in which the final decision is generated by the strongest coalition of ensembles is used. The paper presents preliminary experimental results, which show that the proposed approach improves the quality of classification in terms of F1 weighted measure, balanced accuracy and accuracy. In future work, more extensive experiments are planned for a larger number of data sets. We also consider using different methods for coalitions formation as well as using different approaches to generate the final decision.

References

[1] Bazan, J. G., Bazan-Socha, S., Bentkowska, U., Gałka, W., Mrukowicz, M., and Zaręba, L. Aggregation functions in researching connections between bio-markers and dna micro-arrays. In *International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets*, pages 106–115. Springer, 2022.

[2] Połap, D. and Woźniak, M. A hybridization of distributed policy and heuristic augmentation for improving federated learning approach. *Neural Networks*, 146:130–140, 2022.

[3] Rudin, C., Schapire, R. E., and Daubechies, I. Precise statements of convergence for adaboost and arc-gv. *Contemporary Mathematics*, 443:131, 2007.

[4] Abellan, J., Mantas, C. J., Castellano, J. G., and Moral-Garcia, S. Increasing diversity in random forest learning algorithm via imprecise probabilities. *Expert Systems with Applications*, 97:228–243, 2018.

[5] Hipel, K. W., Fang, L., and Kilgour, D. M. The graph model for conflict resolution: Reflections on three decades of development. *Group. Decis. Negot.*, 29:11–60, 2020.

[6] Zeng, Y., Li, J., Cai, Y., Tan, Q., and Dai, C. A hybrid game theory and mathematical programming model for solving trans-boundary water conflicts. *Journal of Hydrology*, 570:666–681, 2019.

[7] Giordano, R., Passarella, G., Uricchio, V., and Vurro, M. Integrating conflict analysis and consensus reaching in a decision support system for water resource management. *Journal of environmental management*, 84(2):213–228, 2007.

[8] Pawlak, Z. An inquiry into anatomy of conflicts. *Inf. Sci.*, 109(1-4):65–78, 1998.

[9] Pawlak, Z. Some remarks on conflict analysis. *Eur. J. Oper. Res.*, 166(3):649–654, 2005.

[10] Dutta, S. and Skowron, A. Interactive granular computing model for intelligent systems. In *Intelligence Science III: 4th IFIP TC 12 International Conference, ICIS 2020, Durgapur, India, February 24–27, 2021, Revised Selected Papers 4*, pages 37–48. Springer, 2021.

[11] Dutta, S. and Skowron, A. Toward a computing model dealing with complex phenomena: interactive granular computing. In *13th International Conference, ICCCI 2021, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13*, pages 199–214. Springer, 2021.

[12] Liu, D., Liang, D., and Wang, C. A novel three-way decision model based on incomplete information system. *Knowl. Based. Syst.*, 91:32–45, 2016.

[13] Yao, J., Medina, J., Zhang, Y., and Ślęzak, D. Formal concept analysis, rough sets, and three-way decisions, 2022.

[14] Fan, Y., Qi, J., and Wei, L. A conflict analysis model based on three-way decisions. In *IJCRS 2018, Quy Nhon, Vietnam, August 20-24, 2018, Proceedings 6*, pages 522–532. Springer, 2018.

[15] Lang, G., Luo, J., and Yao, Y. Three-way conflict analysis: a unification of models based on rough sets and formal concept analysis. *Knowl. Based. Syst.*, 194:105556, 2020.

[16] Lang, G., Miao, D., and Cai, M. Three-way decision approaches to conflict analysis using decision-theoretic rough set theory. *Inf. Sci.*, 406:185–207, 2017.

[17] Luo, J., Hu, M., Lang, G., Yang, X., and Qin, K. Three-way conflict analysis based on alliance and conflict functions. *Inf. Sci.*, 594:322–359, 2022.

[18] Przybyła-Kasperek, M. and Wakulicz-Deja, A. Global decision-making in multi-agent decision-making system with dynamically generated disjoint clusters. *Applied Soft Computing*, 40:603–615, 2016.

[19] Przybyła-Kasperek, M. and Wakulicz-Deja, A. A dispersed decision-making system–the use of negotiations during the dynamic generation of a system's structure. *Information Sciences*, 288:194–219, 2014.

[20] Aning, S. and Przybyła-Kasperek, M. Comparative study of twoing and entropy criterion for decision tree classification of dispersed data. *Procedia Computer Science*, 207:2434–2443, 2022.

[21] Przybyła-Kasperek, M. and Aning, S. Bagging and single decision tree approaches to dispersed data. In *International Conference on Computational Science*, pages 420–427. Springer, 2021.

[22] Zwitter, M. and Soklic, M. Lymphography. uci machine learning repository, 1988.

[23] Zwitter, M. and Soklic, M. Primary tumor. uci machine learning repository, 1988.

Experimental study of energy savings for TinyML-based multi-gas detection systems

Tobiasz Puślecki[1] [0000−0002−4665−3301]
Krzysztof Walkowiak[1] [0000−0003−1686−3110]

[1] *Wrocław University of Science and Technology*
Department of Systems and Computer Networks
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{tobiasz.puslecki, krzysztof.walkowiak}@pwr.edu.pl

Abstract.
Gas detection is a crucial aspect of safety. The detection systems trigger an alarm based on the type of gas and its concentration in the air. In this article, we present the concept of a TinyML energy-saving system that classifies gases according to their toxicity. We examine the use case with various machine learning (ML) models. Our research demonstrates that the proposed method enhances the TinyML system's runtime performance without compromising the reliability of the gas detection system.
Keywords: *TinyML, energy consumption, gas detection*

1. Introduction

Gas detection is the process of identifying and measuring the presence of various gases in the air. It is an important aspect of safety and environmental monitoring in various industries and settings. The primary goal of gas detection is to prevent or mitigate potential hazards associated with the release of harmful gases.

TinyML refers to the rapidly expanding domain of machine learning technologies and applications designed for constrained resources devices. This includes hardware, algorithms, and software capable of performing on-device sensor data analytics at extremely low power, typically in the mW range and below. Consequently, TinyML facilitates a range of always-on use-cases and is specifically targeted for battery-operated devices [1] [2].

Our work addresses the problem of gas detection with the TinyML device. In [3], the authors adopt a machine learning approach, specifically an ensemble of classifiers, to cope with sensor drift. For this particular application, they choose the Support Vector Machine (SVM) as the base classifier in the ensemble. In [4], the authors present a TinyML-based system for detecting leaks of dangerous gases such as ammonia. The authors of [5] focus on the problem of a recognition system between isooctane and pentane gases based on machine learning analysis techniques, receiving reasonable results for SVM method. The authors of [6] proposed a prototype kitchen monitoring system with gas detection as one of its functions, emphasizing the importance of gas detection in the home environment. Thus, the problem of an energy-efficient gas detection system in which energy-saving parameters can be controlled remains worth developing.

In this article, we introduce a novel concept: a TinyML energy-saving system for ranking gases by toxicity. We evaluate different ML models for this purpose. Our main contribution lies in a new approach to gas detection, particularly suited for battery-powered sensors in hard-to-reach locations. This approach minimizes battery changes while maintaining satisfactory performance.

The rest of the paper is organized as follows: Section 2 describes the conducted experiments, and Section 3 concludes the work.

2. Experiments

The use case under consideration is the detection of harmful gases (vapors) in the air [7]. This dataset contains 13910 measurements (each contains 128 numerical attributes and target variable) from 16 chemical sensors exposed to 6 different gases at various concentration levels. We propose ordering the binary classifiers from the most dangerous to the least dangerous substance, considering the varying toxicity of substances. Each binary classifier represents one substance. Determining this order is non-trivial, considering different states of concentration, symptoms, and density. However, one can attempt to deduce the toxicity based on catalog notes, especially NFPA 704 [8]. Table 1 presents the proposed order of substances according to toxicity.

We use the One-vs-All (*OVA*) strategy, which involves dividing a multi-class classification problem into multiple binary classification problems, one for each class. The classical approach requires recalculating all classifiers (in this case, six). Subsequently, a prediction is generated based on the value of the support function.

Table 1. Proposed order of substances according to toxicity

No.	Name	No. of samples
1.	Ammonia [9]	1641
2.	Ethylene [10]	2926
3.	Acetaldehyde [11]	1936
4.	Toluene [12]	1833
5.	Ethanol [13]	2565
6.	Acetone [14]	3009

In this work, we propose a new approach in which you can set the number of binary classifiers converted in each iteration. Given that not all substances are equally toxic, less toxic substances may be detected less frequently. Detection systems assess air samples at a frequency on the order of hertz. With such rapid inference, delaying the alarm slightly will not cause any damage.

The method presented here involves checking the k-most toxic substances in an iteration with the possibility of cyclically changing the k value in consecutive iterations. Using a 3-tuple like (k_1, k_2, k_3), the value of k in consecutive iterations will take the form $(k_1, k_2, k_3, k_1, k_2, k_3, \dots)$.

In other words, with a 2-tuple like (5, 2), the system will check 5 of the most harmful substances and then 2 of the most harmful substances cyclically in successive iterations. Assuming a sampling rate of 1Hz, the 2 most harmful gases are checked every second, and the 5 most harmful gases are checked every other second. In the classical approach, each gas is checked every second. As described above, some substances can be detected at a lower frequency, so using the k value in the system can save energy by not recalculating all classifiers in each iteration.

After some preliminary experiments, we decided to use the *SVC* method and *StandardScaler* from the scikit-learn library [15]. The *SVC* method performs well with nonlinear and high-dimensional data. The algorithm aims to maximize the decision margin (the distance between the hyperplane and the nearest examples of each class), which can lead to better model generalization. *StandardScaler* is a preprocessing technique used to standardize the features of a dataset and it is important because many machine learning algorithms, such as SVMs, are sensitive to the scale of the input features. The presented concept can be used with any binary classifiers using *OVA* strategies.

We employed the **GridSearchCV** method to identify the optimal hyperparam-

eter combination. Table 2 presents the values of the tested hyperparameters. The dataset was divided into 5 folds in stratified cross-validation. The highest accuracy was achieved with *C=100* and *kernel='rbf'*. Table 3 presents the accuracy metrics for individual binary classifiers and the overall accuracy of all classifiers.

Table 2. The values of hyperparameters for **GridSearchCV** method

Hyperparameter	Values
C	[0.1, 1, 10, 100],
gamma	['auto']
kernel	['linear', 'rbf', 'poly']

Table 3. The values of accuracy metrics for *C=100* and *kernel='rbf'*

Name	Accuracy [%] (std)
Ammonia	98.11 (0.0044)
Ethylene	99.45 (0.0027)
Acetaldehyde	99.64 (0.0031)
Toluene	99.40 (0.0032)
Ethanol	99.84 (0.0008)
Acetone	99.63 (0.0029)
Overall:	99.43 (0.0004)

In the case under study, the binary classifiers have an equal number of parameters, so it can be assumed that a classifier consumes E energy, and k classifiers consume $k*E$ energy. Thus, processing three of the six classifiers saves 50% of energy in the inference process compared to the classical approach. The tuple of subsequent k values is set by the operator. Table 4 presents example k-tuple values and the corresponding inference energy and operating time (omitting sensor energy consumption) according to the classical approach (checking all gases in each iteration). The value of N denotes the number of measurement cycles.

Table 4. Example tuple values with possible inference energy and operating time

***k*-tuple**	**Formula**	**Inference Energy**	**Oper. Time**
Classical	N*6	100%	T
(6,2)	1/2*N*(6+2)	66.66%	1.5T
(6,3,1)	1/3*N*(6+3+1)	55.55%	1.8T
$(k_1, \ldots, k_n)$	1/n*N*($\Sigma_{i=1}^{n} k_i$)	E=(1/n*($\Sigma_{i=1}^{n} k_i$)/6)*100 %	1/E*T

3. Conclusions

In this paper, we have presented an approach that allows us to maintain very similar performance while gaining additional operating time. The use of different k values enables us to explore a tradeoff between accuracy and energy consumption. The method presented here can enhance the TinyML system in terms of operating time at the cost of slightly reduced accuracy.

In future work, we plan to extend the presented concept by incorporating battery level dependency and integrating energy harvesting using solar panels. This will enable the system to recharge batteries and dynamically adjust the k-value in real-time. Further work may also involve exploring additional use cases.

References

[1] Abadade, e. a., Y. A comprehensive survey on tinyml. *IEEE Access*, 11:96892–96922, 2023.

[2] Tsoukas, *et al.*., V. Internet of things challenges and the emerging technology of tinyml. In *2023 19th DCOSS-IoT*, pages 491–495. 2023.

[3] Vergara, e. a., A. Gas sensor drift mitigation using classifier ensembles. *Proceedings of the ACM SIGKDD*, 2011. doi:10.1145/2003653.2003655.

[4] Gkogkidis, e. a., A. A tinyml-based system for gas leakage detection. In *MOCAST 2022*, pages 1–5. 2022. doi:10.1109/MOCAST54814.2022.9837510.

[5] Huichaqueo, e. a., M. Real-time isooctane and pentane gas identification based on machine learning analysis techniques. In *2021 IEEE CHILECON*, pages 1–5. 2021. doi:10.1109/CHILECON54041.2021.9702957.

[6] Ramesh, e. a., A. Home monitoring system using internet of things. In *2023 ICAIS*, pages 1–5. 2023. doi:10.1109/ICAIS56108.2023.10073847.

[7] Vergara, A. Gas Sensor Array Drift Dataset at Different Concentrations. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5MK6M.

[8] National Fire Protection Association. NFPA 704 standard system for the identification of the hazards of materials for emergency response, 2020.

[9] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/4860`. Accessed: 2023-12-29.

[10] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/8655`. Accessed: 2023-12-29.

[11] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/2269`. Accessed: 2023-12-29.

[12] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/4654`. Accessed: 2023-12-29.

[13] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/667`. Accessed: 2023-12-29.

[14] Chemical datasheet. `https://cameochemicals.noaa.gov/chemical/8`. Accessed: 2023-12-29.

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

Digital polarisation: Analysis of 'us' versus 'them' rhetoric in public discussions on social media

Ewelina Gajewska [0009−0006−6012−4787],
Katarzyna Budzynska [0000−0001−9674−9902]

Laboratory of The New Ethos
Warsaw University of Technology
ewelina.gajewska.dokt@pw.edu.pl
katarzyna.budzynska@pw.edu.pl

Abstract. *This work proposes an analysis of social polarisation in the digital environment (online social media) by using a combination of concepts from social science and Natural Language Processing techniques from computer science. Studies show that detection of complex social phenomena such as emotions is still a challenging task even for state-of-the-art architectures [1]. To fill this gap, we conduct an analysis of language of polarisation on Twitter in the debate on climate change using classical concepts from the Aristotle's rhetoric and the power of machine learning tools.*
Keywords: *digital rhetoric, analytics, machine learning, data analysis, social media, social networks*

1. Introduction

Online social networking sites introduced a new dimension of interpersonal communication – global, anonymous and effortless. On the other hand, the design of new media produced new problems – anonymous environment of social media removed the feeling of responsibility for one's own words, thus, the problem of hostile behaviour emerged.

At the same time, the effect of echo chambers – high affinity of like-minded individuals into closed clusters – was observed and claimed as root causes of polarisation. However, further analysis of polarity of these interactions support previous

DOI:10.17388/WUT.2024.0002.MiNI

findings in social sciences – users tend to form tightly connected clusters centred around negative as well as positive references [2]. This is a well-known effect of in-group favouritism and out-group hostility [3].

Polarisation of society goes beyond political divisions and conflicts of opinions - there is a societal problem acknowledged by citizens, which, however, is differently conceptualised by different communities endorsing own sets of values and beliefs regarding the social order. Each of these groups has also their own leaders, authorities, praised by the group members, as well as folk devils, i.e. individuals, groups or organisations blamed for the current problem in society [4].

In this work we provide an analysis of actors perceived by the public as as authorities and devils in public discussions on climate change taking place on Twitter. We propose to analyse these discussions using the frame of Aristotelian rhetoric [5], specifically the concept of ethos adopted by [6] as either attacks or supports of a person's character. We adopt its annotation scheme originally developed on material of parliamentary discussions in UK to specificities of the social media environment. Then, we train RoBERTa model for a ternary recognition of ethotic expressions in natural language.

In order to investigate polarisation in social media, i.e., 'us' versus 'them' rhetoric in public discussions, we use a combination of machine learning and statistical methodologies. A log-likelihood analysis is conducted on top of results returned from the BERT-based model to identify linguistic markers of polarisation. In addition, we make use of graph theory to analyse debate dynamic, that is patterns of clustering of Twitter users (i.e., formation of polarised communities) in discussions on climate change. Here, previous studies observed echo chambers in the anti-vaccination debate on some online social media platforms but not others using the political leaning of news sites quoted by users [7].

2. Methodology

2.1. Data collection

We collect a sample data from Twitter the topic of climate change, specifically two-week span surrounding the Conference of the Parties (COP) in 2019 (COP25), specifically Greta Thunberg's speech at the conference. Discussions on Twitter are collected using a keyword search. We extract 19,171 tweets in total - mentions of Greta Thunberg (both her name and her Twitter account) and replies to Greta Thunberg's posts. Regarding COP conferences, previous studies observed

a pronounced polarisation between supporters and contrarians of the mainstream climate change rhetoric [8].

2.2. Language model

In order to analyse polarity of interactions and attitudes of speakers towards each other, we make use of Aristotelian concept of ethos operationalised in [6] as attack and support ethotic expressions, i.e., a favourable or unfavourable reference to a speaker. In this work we also extend previous studies on automatic detection of ethotic expressions in parliamentary speeches by professional politicians - we investigate the use of rhetorical strategies of appeals to ethos in online discussion by laypersons.

We fine-tune RoBERTa model, available in the python Transformers library, for automatic recognition of attack and support (versus neutral) expressions in natural language. It was trained for 2 epochs using Pytorch with 5e-5 learning rate and dedicated function from the Transformers library. The model extends previous tool that relied on a two-step detection as well as extensive set of lexical features (part of speech and dependency tags, entity extraction, polarity). Its performance is satisfactory - the model achieves 0.758 macro $F1$ score on a test set (over 2 times higher result compared to two baseline classifiers using most frequent class label (0.223 $F1$ score) and empirical class prior probabilities (0.328)). Thus, results returned by the model are used for statistical analysis.

2.3. Statistical analysis

We lemmatise text and apply log-likelihood calculation according to the formula by University Centre for Computer Corpus Research on Language of Lancaster University[1]. We calculate odds ratio of words being classified as part of sentences that support or attack someone's ethos. The test was previously employed to identify toxicity triggers in online communities on Reddit [9].

3. Results

Results of statistical analysis of language of polarisation are presented in Table 1. Offensive words such as 'bullshit', 'dumb' and 'clowns' are employed by social media users to attack Thunberg and other individuals and groups. The word

[1] https://ucrel.lancs.ac.uk/llwizard.html

'liar' on top of the table suggest that individuals are most likely to be attacked for lying. We identify disagreement with another speaker, marked by the use of words 'hypocrite', 'clowns' and 'bullshit', is another motive to personally attack other speakers. Positive words describing someone's character ('bravery', 'fearless', 'wise') as well as verbs communicating praise ('admire', 'proud') in turn are used to support the character of others. Expression of gratefulness ('thank', 'grateful') also fall into this category. Results of our lexical analysis could be regarded as extending previous findings on the phenomenon of in-group favouritism and out-group hostility observed in psychological experiments [10].

Table 1: Language of polarisation based on results from the BERT classification of ethotic expressions

Rank	Word	Odds	Appeal	Word	Odds	Appeal
1	liar	68.91	attack	admire	65.43	support
2	gurus	46.76	attack	bravery	58.65	support
3	marxist	44.3	attack	grace	40.6	support
4	hypocrite	40.61	attack	fearless	40.6	support
5	mentally	39.37	attack	inspiration	33.89	support
6	globalists	39.37	attack	proud	27.35	support
7	dumb	35.07	attack	thank	17.77	support
8	clowns	34.45	attack	amazing	15.28	support
9	liberals	33.22	attack	wise	15.23	support
10	bullshit	30.15	attack	grateful	14.29	support

With the use of our method we also identify entities attacked by the public in social media, that we therefore regard as folk devils. In the context of climate change and Thunberg's speech liberals and globalists are regarded as such on Twitter. In addition, in Figure 1 we present names of individuals that are most frequently attacked in these tweets. Both regular Twitter accounts and well-known politicians (e.g. Joe Biden) and activists (e.g. Leonardo DiCaprio) are attacked for their actions, words or decisions regarding climate issues.

Finally, we analyse a social network of interactions on Twitter. With the use of ethos detection techniques, we observe similarly to [2] that users form both negative as well as positive interactions between users (clusters, debate communities). We suspect these references correspond to interactions with members of the out-group and members of the in-group, respectively. Such an analysis of polarity of

Figure 1: Negative actors in the debate on climate change on Twitter.

interactions allows to better observe dynamic of online discussion (see left side of Figure 2) than a sole analysis of user mentions (right side of Figure 2).

Here we categories both each interaction between users (graph edges) and users themselves (graph nodes) as attacking, supporting, ambivalent and neutral. Labels returned by our model constitute categories for interactions (i.e., tweets that mention other individuals attack, support or neutrally reference other individuals). Then, a user is categories as attacking (supporting) if he performs only attacks (supports); ambivalent users (marked by yellow colour in Figure 2) both attack some persons and support others, neutral users do not use ethotic expressions.

4. Conclusions

This work proposes a method of analysis of polarisation, specifically its linguistic dimension, using the Aristotelian concept of rhetoric. Instead of focusing on discourse analysis well-known in linguistic studies we use the power of machine learning to investigate polarisation (its linguistic dimension as well as network dynamics) in the digital environment of social media.

Such large-scale studies would not have been possible with the use of sociological surveys of public opinion or manual analyses of online content by expert annotators (e.g., linguists, psychologists). Although the phenomenon of polarisa-

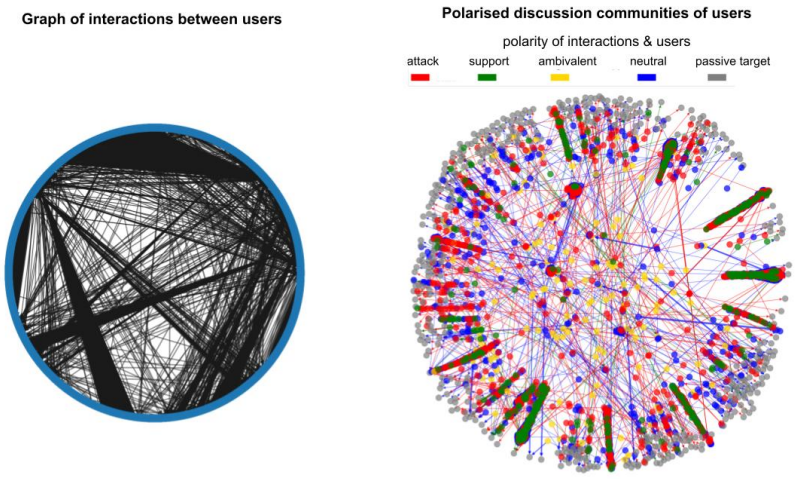


Figure 2: Network analysis of discussion on Twitter. Graph of interactions (replies) between users in online debate gives a scant picture of the debate dynamic. Analysis of polarity of these interactions (graph edges) along with categorisation of users (nodes) as attacking, supporting, ambivalent or neutral gives a more detailed illustration of this dynamic.

tion has been studied thoroughly in social science, its methods are insufficient to address the problem at a large scale in the current state of digital communication [11].

Acknowledgment

The work reported in this paper was supported in part by CHIST-ERA under grant 2022/04/Y/ST6/00001.

References

[1] Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Bielaniewicz, J., Gruza, M., Janz, A., Kanclerz, K., et al. Chatgpt: Jack of all trades, master of none. *Information Fusion*, page 101861, 2023.

[2] Keuchenius, A., Törnberg, P., and Uitermark, J. Why it is important to consider negative ties when studying polarized debates: A signed network analysis of a dutch cultural controversy on twitter. *PloS one*, 16(8):e0256696, 2021.

[3] Bliuc, A.-M., Bouguettaya, A., and Felise, K. D. Online intergroup polarization across political fault lines: An integrative review. *Frontiers in Psychology*, 12:641215, 2021.

[4] Zielińska, I. and Pasamonik, B. Polarizing moral panics: A theory and its application to the refugee crisis in poland. *Deviant Behavior*, 43(12):1543–1557, 2022.

[5] Aristotle. *The Art of Rhetoric*. Penguin Books, London, U.K., 1991.

[6] Duthie, R. and Budzynska, K. A deep modular rnn approach for ethos mining. In *IJCAI*, pages 4041–4047. 2018.

[7] Cinelli, M., Morales, G. D. F., Galeazzi, A., Quattrociocchi, W., and Starnini, M. Echo chambers on social media: A comparative analysis. *arXiv preprint arXiv:2004.09603*, 2020.

[8] Falkenberg, M., Galeazzi, A., Torricelli, M., Di Marco, N., Larosa, F., Sas, M., Mekacher, A., Pearce, W., Zollo, F., Quattrociocchi, W., et al. Growing polarization around climate change on social media. *Nature Climate Change*, pages 1–8, 2022.

[9] Almerekhi, H., Kwak, H., Salminen, J., and Jansen, B. J. Provoke: Toxicity trigger detection in conversations from the top 100 subreddits. *Data and Information Management*, 6(4):100019, 2022.

[10] Tajfel, H., Billig, M. G., Bundy, R. P., and Flament, C. Social categorization and intergroup behaviour. *European journal of social psychology*, 1(2):149–178, 1971.

[11] Zhang, J., Wang, W., Xia, F., Lin, Y.-R., and Tong, H. Data-driven computational social science: A survey. *Big Data Research*, 21:100145, 2020.

Rephrase Analytics for Digital Rhetoric: Exploring The Dynamics of Reformulation in Natural Dialogue

Maciej Uberna[0009−0006−8953−8270],
Katarzyna Budzynska[0000−0001−9674−9902],
Marcin Koszowy[0000−0001−5553−7428]

Laboratory of The New Ethos:
(https://newethos.org/technologies/)
Faculty of Administration and Social Sciences
Warsaw University of Technology

Abstract. *In this paper, we propose to develop analytics for natural language data containing rephrase – a communication phenomenon that consists of reformulating a message in order to achieve a variety of persuasive goals. In order to establish a tool for analysing and visualising statistical data about the dynamic rephrase uses, both qualitative and quantitative methods have been employed to study annotated corpora for rephrase. As a result, a novel speech analytics technology has been developed that helps users identify and visualise various aspects of this phenomenon for the sake of the development of digital rhetoric.*
Keywords: *rephrase, visualization of linguistic data, artificial intelligence, data mining, qualitative analysis, quantitative analysis, corpus studies*

1. Introduction

This paper touches upon the issue of the lack of unified account of the three basic aspects of communication in digital rhetoric: *logos* (communication structures of an inferential-argumentative nature), *ethos* (structures having to do with the credibility, social position and authority of the participants in the communication process) and *pathos* (communication structures containing appeals to the emotions of the recipients). Despite the fact that each of these aspects is analyzed

DOI:10.17388/WUT.2024.0002.MiNI

within the framework of numerous theoretical and implementation models and approaches [1], no unified model has yet been developed to identify all these three aspects with a particular focus on identifying the relationships between them.

In what follows, we propose to develop a technology of digital rhetoric for one aspect of rhetoric analytics, namely rephrase, a reformulation of a message that is aimed for achieving a variety of rhetorical gains, such as, e.g. strengthening the persuasive force of a message or emphasising a certain aspect of a raised issue for the sake of better comprehension by an audience [2]. In terms of communication structures, the rephrase relation holds between the rephrase input (an original message) and the rephrase output (a reformulated piece of information). The Rephrase Analytics tool proposed in this paper aims at exploring the dynamics of rephrase in the annotated corpora for (i) US2016 Presidental Election debates, (ii) social media (Reddit) users' reactions to those debates, and (iii) face-to-face UK parliamentary debates. The preliminary results show how to systematically explore the *logos-ethos-pathos* interface for the sake of developing digital rhetoric.

2. State of the art

Extracting data from discourse, making sense of them and displaying them in a readable way is the subject-matter of research conducted within a variety of disciplines, including (i) argumentation theory [1] with a special focus on computational models of argumentation and theoretical models and computer tools for representing complex communication structures, (ii) rhetoric with a focus on digital rhetoric [3] and rhetorical structure theory [4], and (iii) Inference Anchoring Theory (IAT) that combines logical, dialogical and pragmatic aspects of communication [5] as well as its computational counterpart, the Argument Interchange Format ontology [6].

Among the increasingly sophisticated theoretical and methodological attempts to capture the relationship between *logos*, *ethos* and *pathos* as key aspects of communication, the most noteworthy is the Inference Anchoring Theory as a modeling framework to capture the relationship between inferential-argumentative structures (*logos*), selected "ethotic arguments" referring to the character and authority of the speakers (*ethos*) and selected arguments aimed at arousing emotions among the audience (*pathos*). This theory, while having a conceptual scheme to identify the complex structures of *logos*, only partially captures some detailed structures of *ethos* and *pathos*, and as such does not provide a systematic account of the

relationship between the three types of communication structures.

In 2021, there were partially successful attempts to extract the basic components of *logos* (relations of inference, conflict and rephrase) from text corpora using transformer-based neural networks [7], and in 2023, more accurate methods for segmenting corpora into text units were developed [8] enabling more accurate automatic annotation of corpora. Nevertheless, a research gap is apparent, for example, in the case of the latest methods for annotating argumentation patterns [9] using IAT, which allow only for a partial identification of *ethos* and *pathos* structures [10]. This paper is aimed at filling in this gap by proposing digital rhetoric technology for rephrase as one particular aspect of the communication. By combining the applications of computerized tools for representing, processing and visualizing natural language data with the methods of corpus linguistics, argumentation theory and rhetoric, this tool would comprehensively capture the main types of communication strategies related to rephrase uses.

3. Data, methods and results

3.1. Annotated corpora with instances of rephrasing

The work reported in this paper consists of implementing the prototype of Rephrase Analytics to three corpora of arguments annotated with OVA – *Online Visualisation of Arguments* (`http://ova.arg-tech.org/`) – a software that allows for mapping communication structures identified by the Inference Anchoring Theory [5, 10]. This tool is applied for exploring the dynamics of rephrase in the annotated corpora for (i) US2016 Presidential Election debates (58,900 words), (ii) Reddit users' reactions to those debates (39,099 words), and (iii) face-to-face UK parliamentary debates (90,000 words)[1].

3.2. Data pre-processing and processing

The data generation process started with a manual annotation of arguments and rephrase structures using the OVA tool. Also, the manual annotation of appeals to ethos (i.e. speaker's charisma and character) on top of argumentation structures has been added. The ethos annotation took into account three categories: E+ –

[1]For the annotated corpora stored in the Argument Interchange format Database, see `https://corpora.aifdb.org/US2016`, `https://corpora.aifdb.org/IATHansard1`, and `https://corpora.aifdb.org/IATHansard2`.

for an ethos support (e.g. "Trump is going to cut taxes big league"), E- – for an attack on ethos (e.g. "Clinton doesn't have the stamina"), and E0 – for no ethos. Next, the sentiment (i.e. speakers' expressed emotions) was mined automatically from argumentative structures present in the corpora, using the package called "transformers[sentencepiece]" executed in Python in the Google Colab environment. Three sentiment-related categories have been distinguished: S+ for the positive sentiment, S- for the negative emotional load, and S0 for sentiment-neutral expressions. Finally, the dynamics of ethos and sentiment was captured by analysing rephrase inputs and outputs. For example, rephrasing an input "I recently heard this story" to the output "I recently heard this thrilling story" relies on adding positive sentiment (S+) to the rephrase output which contained no sentiment (S0). This kind of rephrase dynamics was called "Amelioration" in order to emphasise the discourse dynamics the result of which is a positive sentiment expression. Apart from the Amelioration, two other types of rephrase dynamics were distinguished, namely "Pejorativization" (a transition from S0 or S+ to the negative sentiment expression, S-), and "Neutralization" (a change from S+ or S- to a sentiment-neutral expression, S0).

The information regarding all these dynamic transitions can be loaded from Excel file to the Rephrase Analytics as separate corpora that can dynamically be merged by a user with various combinations and configurations. For example, we can select only two categories such as "Amelioration" and "Neutralization" for selected corpora, such as the UK parliamentary debates and the US 2016 Presidential Election debates. For the purpose of data handling and display, several libraries have been used, such as (i) Streamlit for graphical interface and Winsocket handling, (ii) Pandas and NumPy libraries for data manipulation, (iii) WordCloud for word statistics, and (iv) Plotly library for 2D and 3D data visualization.

3.3. Quantitative and qualitative methods for Rephrase Analytics

Qualitative analysis employed in Rephrase Analytics helps us to identify ways of using rephrase to e.g. strengthen or emphasize the arguments used or to misrepresent the opponent's claims by e.g. oversimplifying or exaggerating an opponent's standpoint [1, p. 550]. Additionally, rephrase can be used in the opposite way to 'soften arguments' making them more friendly for the audience. The analysis of all these aspects requires a qualitative and quantitative approach.

In terms of the quantitative analysis, the analyst's interface allows for selecting the most frequent rephrase instances that affect the ethos or expressed emotions of

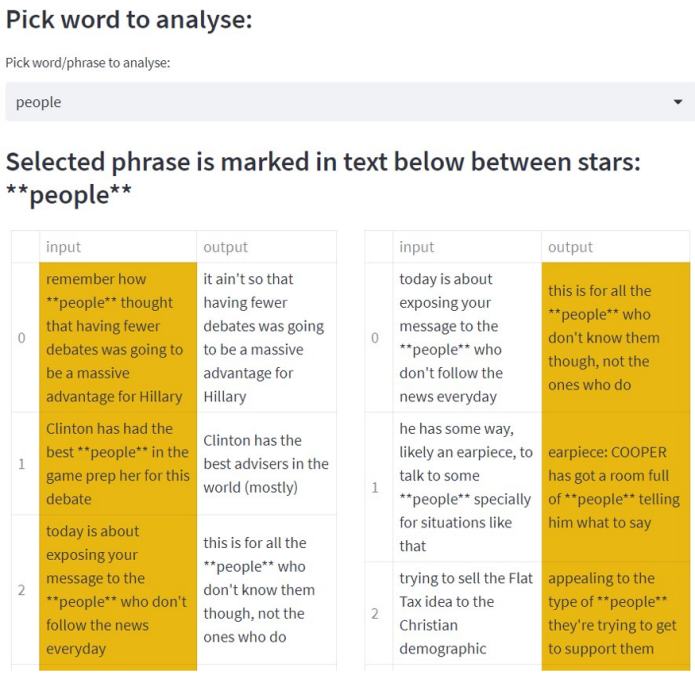


Figure 1: Data visualization - qualitative analysis

the speaker, then filters allow to select whether the rephrase strengthens, weakens or neutralizes the first statement (*input*), and then in qualitative module certain word phrases can be selected from the discourse, divided into (*input*), (*output*) or both in order to see the usage and function of rephrase in the discourse.

3.4. Preliminary findings

Figures 1 and 2 depict pre-selected forms of data visualization of ethos or sentiment (*speakers' expressed emotions*) dynamics in rephrase that correspond to qualitative (1) and quantitative (2) analysis.

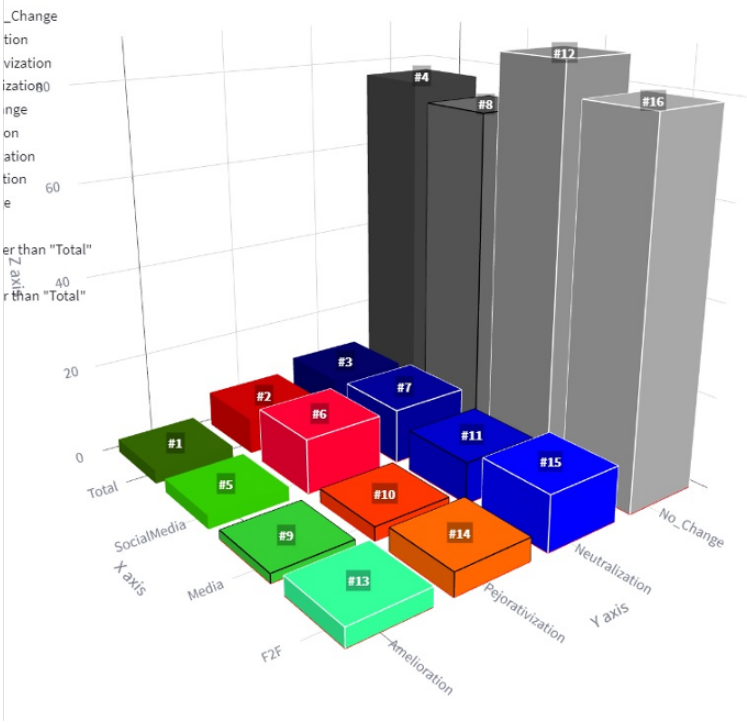


Figure 2: 3D data visualization of rephrase aspects

Among other features such as word clouds the Rephrase Analytics tool helps us to identify single words or phrases that are typical for either rephrase input or rephrase output (see figure 1), that is useful for an analyst to explore linguistic cues for most frequent persuasive rephrase uses.

As long as quantitative analysis is concerned, Rephrase Analytics is helpful in doing the cross-corpora analysis that shows how frequent in the data are the instances of rephrase that e.g. add an attack on other speaker's ethos in the rephrase output or add in the output an additional emotional load, as in the example from the first US 2016 Presidential Debate at Hofstra University (September 26, 2016), when Donald Trump, after addressing Hillary Clinton with "You are going to drive businesses out" rephrased it then to "You are going to regulate these businesses out of existence". For instance, as figure 2 shows, in social media (Reddit) the most frequent dynamic change is negative (*pejorativization* - change from positive/neutral to negative) for both ethos and sentiment in the rephrase. This result indicates that the social media users want in general to repeat their statements and the second most frequent strategy is to attack others.

4. Conclusion and future work

The Rephrase Analytics tool, by combining the accounts of argumentation, speaker's character, and expressed sentiment within a unified model frame to capture a variety of rephrase uses in natural dialogues, helps us process and visualize statistical data on the identified communication phenomena in the corpora using quantitative and qualitative methods of discourse analysis (including methods of corpus linguistics, argumentation theory and Inference Anchoring Theory). Given recent experimental research proving that rephrase is far from being a marginal phenomenon in communication [2], the further development of Rephrase Analytics may help statistically obtain the knowledge of most typical rephrase uses, and, once some of them have been identified as misuses of rephrasing, may become useful for developing automatic moderation and mediation software.

5. Acknowledgements

The work reported in this paper was supported by the Polish National Science Centre under grant 2020/39/I/HS1/02861.

References

[1] van Eemeren, F., Garssen, B., Krabbe, E., Snoeck-Henkemans, A., Verheij, B., and Wagemans, J. *Handbook of argumentation theory*. New York: Springer, 2014.

[2] Younis, R., de Oliveira Fernandes, D., Gygax, P., Koszowy, M., and Oswald, S. Rephrasing is not arguing, but it is still persuasive: An experimental approach to perlocutionary effects of rephrase. *Journal of Pragmatics*, 210:12–23, 2023.

[3] Eyman, D. *Digital Rhetoric: Theory, Method, Practice*. MI: Digital Culture Books, 2015.

[4] Mann, W. and Thompson, S. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[5] Budzynska, K. and Reed, C. Speech Acts of Argumentation: Inference Anchors and Peripheral Cues in Dialogue. In *Computational Models of Natural Argument*. 2011. URL `http://cmna.csc.liv.ac.uk/CMNA11/3940-16678-1-PB.pdf`.

[6] Chesnevar, C., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., and Willmott, S. Towards an argument interchange format. *The Knowledge Engineering Review*, 21(4):293–316, 2006.

[7] Ruiz-Dolz, R., Heras, S., Alemany, J., and García-Fornes, A. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intelligent Systems*, 36:62–70, 2021.

[8] Fan, Y., Li, B., Sataer, Y., Gao, M., Shi, C., Cao, S., and Gao, Z. Hierarchical clause annotation building a clause-level corpus for semantic parsing with complex sentences. *Applied Sciences*, 13:1–24, 2023.

[9] Walton, D., Reed, C., and Macagno, F. *Argumentation Schemes*. Cambridge University Press, New York, 2008.

[10] Visser, J., Lawrence, J., Reed, C., Wagemans, J., and Walton, D. Annotating argument schemes. *Argumentation*, 35:101–139, 2021.

AI-supported Self-Learning Application using Emotion Recognition and Focus Analysis

Kamila Skorupka[0009−0004−8212−5698],
Weronika T. Adrian[0000−0002−1860−6989]

AGH University of Krakow
Department of Applied Computer Science
Al. Mickiewicza 30, 30-059 Kraków, Poland
kskorupka@student.agh.edu.pl, wta@agh.edu.pl

Abstract. *Emotions and focus determine people's ability to learn efficiently. Constant distractions and sub-optimal levels of arousal can be detrimental to one's attempt to learn with computer software. To this end, it is interesting to research how we can analyze and control users' emotions and focus within an AI-supported self-learning application. The focus of this paper is on emotion and focus recognition in a practical setting of a self-learning application using flashcards. We implement a supervisory system that adapts to the recognized user state to optimize their ability to learn efficiently. We present a proof-of-concept system and discuss the obtained results.*
Keywords: *affective computing, emotion recognition, focus analysis, artificial intelligence*

1. Introduction

From the earliest years of our lives, a variety of feelings accompany us, and as time passes, we not only learn to recognize them but also refine the skill of control, using them as a tool to achieve our goals. There is an increasing conversation about the role of emotions in everyday life, in particular in education and learning [1, 2].

The primary motivation behind the research presented in this paper stemmed from the burgeoning trend of using self-education apps (such as Quizlet or Duolingo) and the increasing interest in affective computing within the field of Computer Science. There are numerous projects revolving around affective computing that aim to enhance people's daily lives. One notable example is "The

DOI:10.17388/WUT.2024.0002.MiNI

Little Big e-motion" created by Hyundai – featuring Emotion Adaptive Vehicle Control (EAVC) technology, supports young patients at the SJD Barcelona Children's Hospital by optimizing the vehicle environment based on the driver's mood, to improve the patients' health condition and prepare them for medical examinations. Through technologies like this, we witness how the fusion of emotional intelligence and computing power can positively impact various aspects of our lives, from personal interactions to productivity.

Observing these trends, we aimed to develop an application for self-learning, leveraging flashcards with a supervisory system. Our goal was to enhance the quality of learning within the app while aiding users in maintaining focus, utilizing artificial intelligence algorithms for facial emotion recognition and controlling the level of focus. The main contribution of this work is a demonstration of a prototypical tool that observes, recognizes, and reacts to the user's emotional state and focuses on the task at hand. The application has been designed based on interdisciplinary research in psychology, cognitive science, and artificial intelligence, to ensure that proper requirements are defined and appropriate methods are selected.

The paper is organized as follows: in Section 2, we introduce the basic notions related to emotion analysis, on which we based the design of the application. In Section 3, we outline the algorithms, models, and technologies used in the system. In Section 4, we briefly evaluate the proof-of-concept of our research and we conclude the paper in Section 5.

2. Influence of human's emotions and focus on learning

To this day, scholars debate what exactly the concept of emotions encompasses. Despite various perspectives and approaches, there is some similarity in their understanding – most agree that emotions can be described considering two key aspects: valence, which refers to the direction of emotions from positive to negative, and arousal, which denotes the intensity level of emotional response [3], as depicted in Figure 1.

Thanks to these two dimensions, we can understand the correlation between emotions and the memorization process more clearly. It is worth noting that it is not the emotions themselves that directly influence cognitive processes. The focus here is on *affect*, which is a short-lived organismic response to emotions [4]. Therefore, we can consciously manipulate the reactions that occur in the body thanks to emotions. Here, knowledge supported by the research of R. Yerkes and

J. Dodson is useful, as it is known that there is a correlation between the level of emotional arousal and human performance [5], as depicted in Figure 2.

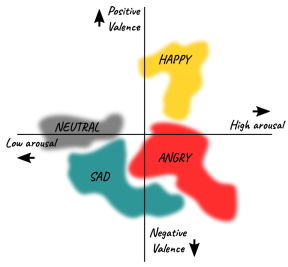


Figure 1. The diagram illustrates a two-dimensional space of emotions.

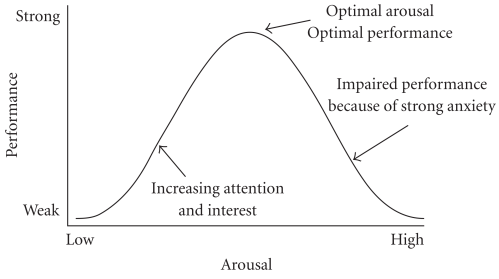


Figure 2. The relationship between arousal and performance based on a modified version of the Yerkes-Dodson law, as adapted by Donald Olding Hebb.

We can observe that humans achieve optimal performance at an optimal arousal level. This state can be achieved through methods of arousal and emotion regulation. Thanks to such observation, we may assume that the best way to maintain high performance while learning is to keep the person in optimal arousal.

3. Self-learning application with a supervisory system

The developed application is a web-based platform that enables users to learn through flashcards. Using the user's camera feed, the application monitors the user to provide assistance during learning sessions and keep them motivated. It ensures user engagement by tracking their focus through screen gaze detection and utilizes emotion regulation techniques to maintain a positive mood, adjusting arousal levels optimally (either lowering or raising them).

At the core of the application lies a supervisory system, segmented into two submodules: one dedicated to sustaining focus, and the other to regulating emotional arousal. Both modules utilize live video feed from the user's camera, captured every half second, to dynamically manage the user's learning session. The

Focus Submodule leverages the FaceMesh[1] machine learning model provided by the MediaPipe framework, enabling it to access a map of the user's facial landmarks, thus facilitating control over the user's level of focus. Additionally, it possesses transformation matrices (for converting points from 2D space to 3D space) and the latest recorded points of the user's iris centers. Using this information, it estimates the user's gaze vectors, and subsequently evaluates whether the user is focused or not. On the other hand, the Emotion Submodule utilizes a Haar cascade classifier [6] to identify faces in the captured image. It also employs an advanced neural network model, which, based on the facial image, generates a response in the form of a probability vector assigned to the predicted dominant emotion class. The artificial neural network model was trained using transfer learning on a dataset from Kaggle[2]. Pre-trained models such as MobileNet2, InceptionResNet2, VGG16, and ResNet50 were utilized during the training process.

Both submodules dynamically impact the progression of the user's learning session according to a *Supervisory System Algorithm* that reads as follows:

Require: *user* is in learning session
Ensure: *application* has access to *user*'s camera
  while learning session is not finished **do**
    if *user* was unfocused for the last 3 seconds **then**
      Send an alert indicating *user* is not focused.
      Clear the focus-emotion *stack*.
    else if *user* experienced only negative emotions for the last 3 seconds **then**
      Send an alert about the issue with emotional arousal.
      Identify the dominant emotion.
      if sadness or disgust **then**
        Increase emotional arousal.
        Clear the focus-emotion *stack*.
      else if anger or fear **then**
        Decrease emotional arousal.
        Clear the focus-emotion *stack*.
      end if
    end if
  end while

[1] https://developers.google.com/mediapipe/solutions/vision/face_landmarker
[2] https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset

4. Results and discussion

A series of experiments was conducted on the application, with a primary focus on the learning session. The application effectively identifies the user's lack of focus, as shown in Figures 3 and 4; however, an issue arises when detecting more than one face, e.g., when two individuals are positioned close to each other.

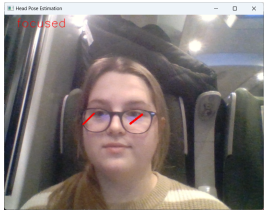


Figure 3. Good focus

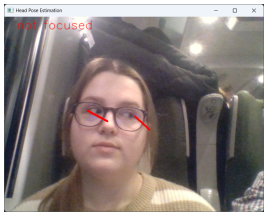


Figure 4. Lack of focus

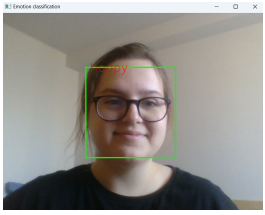


Figure 5. Joy detected

Another issue lies with the neural network model, which does not always accurately recognize sadness or anger – very often it categorizes these emotions as a neutral state. This is likely due to overfitting of the model, as depicted in Figure 6. However, the model performs quite well in recognizing most emotions, as depicted in Figure 5.

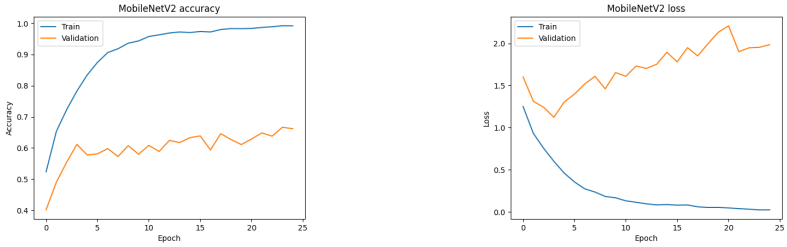


Figure 6. Accuracy and loss curves of the best-trained model.

The application could be enriched by creating a user community, enabling them to compete with each other. It would be valuable in the future to conduct comparative tests on a specific research group to verify the usefulness of the application. Additionally, considering the utilization of IoT devices for more precise emotion verification would be worthwhile.

5. Summary

AI-supported emotion and focus recognition can help develop more efficient applications for self-learning. In this paper, we presented an application for self-education with a supervisory system aimed at enhancing the quality of users' learning. An evaluation of the proposed solution was conducted, focusing on monitoring user engagement and emotional arousal while using the application. It was found out that the application effectively detects the lack of user focus and can recognize their emotions to a greater extent, thereby enabling the environment to support the user. In the future, it would be beneficial to carry out comparative tests on a designated research group to validate the efficacy of the application.

Acknowledgment

The authors would like to thank Paweł Jemioło for his valuable insights and comments on affective computing followed by emotion modeling and recognition.

References

[1] Pekrun, R. *Emotions and learning*, volume 24. International Academy of Education (IAE) Geneva, Switzerland, 2014.

[2] Giżycka, B., Jemioło, P., Domarecki, S., Świder, K., Wiśniewski, M., and Mielczarek, L. A thin light blue line-towards balancing educational and recreational values of serious games. In *AfCAI*. 2019.

[3] Martin, R. E. and Ochsner, K. N. The neuroscience of emotion regulation development: Implications for education. *Current opinion in behavioral sciences*, 10:142–148, 2016.

[4] Niven, K. and Miles, E. *Affect Arousal*, pages 50–52. Springer New York, New York, NY, 2013. ISBN 978-1-4419-1005-9. doi:10.1007/978-1-4419-1005-9_1089. URL https://doi.org/10.1007/978-1-4419-1005-9_1089.

[5] Caprì, T., Fabio, R., Towey, G., and Antonietti, A. *Current Theory*. 2020.

[6] Soo, S. Object detection using haar-cascade classifier. *Institute of Computer Science, University of Tartu*, 2(3):1–12, 2014.

Autoencoder model in anomaly detection for weather data

Aleksandra Grzelik[0000−0002−8380−6469]

University of Łódź
Faculty of Mathematics and Computer Science
Stefana Banacha 22, 90-238 Łódź, Poland
aleksandra.grzelik@edu.uni.lodz.pl

Abstract. *The study covers application of autoencoder model to binary classification and anomaly detection in weather data. The problem explored is automatization of data labelling, which would enable processing of the growing amount of unlabelled data. The architecture of the autoencoder network and search for adapting the model to outlier-prone data are discussed. Model optimization part addresses issues such as model's ability to generalize and to map hidden data structure.*

Keywords: *autoencoder, anomaly detection, automatic labelling, classification*

1. Introduction

The study aims to use autoencoder models for weather data and includes automatic labelling based on binary classification and anomaly detection. The automatic labelling approach addresses the problem of the exponential growth of the available amount of unlabelled data, lack of which limits the scope for analysis. Neural network-based solutions offer relatively high efficiency with minimal workload are therefore an excellent answer to the problem. The non-linearity introduced into the model provides the networks with the ability to learn, which is a significant advantage in processing large data sets.

DOI:10.17388/WUT.2024.0002.MiNI

2. Data description

The study bases on data published by Polish National Institute of Meteorology and Water Management, covering the period from as far back as 1951 to 2023 for Polish meteorological stations [1]. Twelve stations were selected for further analysis: Dynów, Kołuda Wielka, Kraków-observatory, Legionowo, Lidzbark Warmiński, Puczniew, Puławy, Pułtusk, Skierniewice, Szepietowo, Warszawa-Bielany, Wielichowo. Selection was motivated by the availability of historical data. From various options of data time intervals and topics, the selection contains monthly data regarding climate parameters such as variants of temperature, amount of rainfall days or days with snow cover.

3. Autoencoder model

Autoencoder is an artificial neural network model, initially proposed as a generalization of principal component analysis [2], which is appropriate for dimensionality reduction of the data containing nonlinear relations. Various types of autoencoders are used to conduct unsupervised learning and are widely applied to problems such as classification or anomaly detection, mostly on image datasets.

The autoencoder consists of two subnetworks: an encoder and a decoder. The encoder's objective is to compress the structure of the input data, while the decoder acts in reverse, it reconstructs the input layer from the obtained compression. The model learns by minimising loss function which represents the divergence between input and output values, ideally working autoencoder would return input's identity. However, it is important to avoid a situation where the model remembers the input data to return it as output. To prevent this, the hidden layer, a bottleneck, must be of lower dimension than the input and output layer.

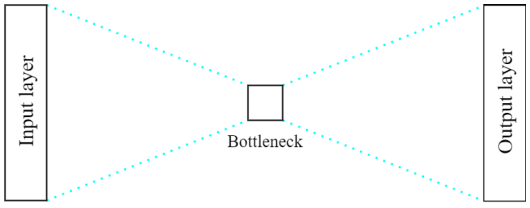


Figure 1. General autoencoder model

Encoder is a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that maps n-dimensional input x to its representation h:

$$h = f(x) = s_1(W_1 x + b_m)$$

where s_1 is a nonlinear activation function, W_1 is a weights matrix and b is an m dimensional bias [3]. The decoder function $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ maps a representation h back to the n dimensional space:

$$y = g(h) = s_2(W_2 x + b_n)$$

where s_2 is an activation function, W_2 represents weights and b is an n dimensional bias vector.

3.1. Search for robustness

When building an autoencoder model, one of the most important choices to be made is a choice of loss function. Two functions that are popular among neural networks' loss functions are Mean Square Error (MSE) and Mean Absolute Error (MAE). Both of which remain unique in their simplicity, but also have certain limitations. MSE is defined by the following formula:

$$\text{MSE} = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}$$

where y_i is an empirical value, $\hat{y}_i$ the value based on model predictions and N is the number of observations. Main goal of MSE is scaling the value of the loss function according to the value of the error – larger errors are penalised far more than small ones. It may be a positive quality, but it increases risk of overfitting and makes model inadequately prone to outliers. Its most popular alternative is Mean Absolute Error, providing higher level of robustness, but is not as effective in minimizing larger errors:

$$\text{MAE} = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}$$

Huber loss function [4] is a combination of absolute value and rectangular functions, which combines the strengths of both MSE and MAE approaches, providing robustness of the model, yet allowing it to learn effectively.

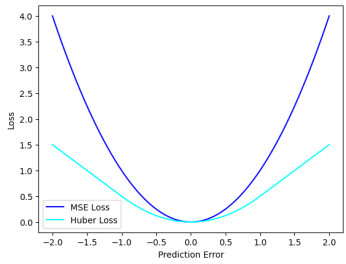


Figure 2. Loss functions visualisation

Huber loss function, given by formula 1, remains linear for errors greater than δ (which reduces overfitting risk), while for errors smaller than δ it allows non-linearity which, depending on the data in question, can lead to more effective learning:

$$L_\delta = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 & \text{for} \quad |y_i - \hat{y}_i| \leq \delta \\ \\ \frac{1}{N}\sum_{i=1}^{N}\delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \tag{1}$$

where δ is a hyperparameter adjusting the threshold for switching from quadratic to linear loss function. Figure 2 represents the difference between MSE and Huber loss approaches.

4. Experiments and results

The experiments were divided into two parts: automatic labelling based on binary classification and anomaly detection. In the first part, the results of a model were compared for different test sets. In the second part, two models, learning on different datasets were analysed.

4.1. Comparison of models

The models: `model_1`, `model_2`, `model_3` were based on the following architecture: a twelve neuron input layer, one hidden layer containing a single neuron – the so-called bottleneck and a twelve-dimensional output layer. The hidden layer stores the encoded values while the output layer presents the decoded values. In

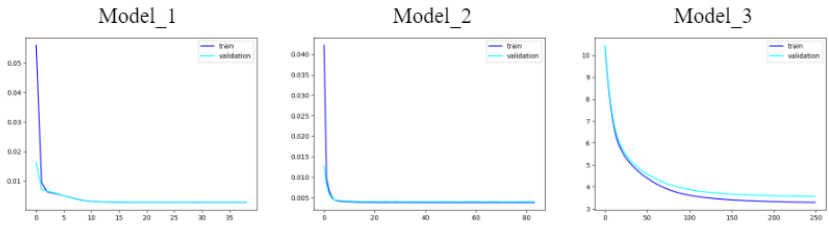


Figure 3. Model training visualisation

each of three models, the encoder layer was activated with a sigmoid function and the decoder layer with a linear function, which was inspired by [3].

The models `model_1` and `model_2` were designed to be used in the anomaly detection section. Training sets for `model_1` and `model_2` consisted of 8387 observations, with and without trend component respectively. Both of these models were tested and validated with 1048 record sets.

The automatic labelling section bases on `model_3`, its training set was modified so that the model would be adequate for performing binary classification of records. The `model_3` was trained on 1600 records set and validated on 400 records. The test set used to verify the performance of `model_3` was divided into three subsets, which consisted of 100, 120, 130 records respectively.

As each learning set was split into training and validation set, it could be concluded that the models have the ability to generalise. The value of the loss function for each model's training and validation set during learning procedure is presented in figure 3.

4.2. Autoencoder in automatic labelling

The model used for automatic labelling of weather data, `model_3`, learns with a dataset based on observations from the winter months: December, January and February. The aim of the model is to identify winter and non-winter observations. The labelling is based on a binary classification of the data from the test sets, which include observations from the winter months (those not present in the training set) and: observations from July (test 1), April, May and June (test 2), August and September (test 3). The table 1 presents the scores of classifier for the test sets. For each of the sets, the model maintains appropriate values of accuracy, precision

measure	test 1	test 2	test 3
Precision	0.952	1.000	0.909
Recall	1.000	0.978	1.000
Accuracy	0.990	0.993	0.973

Table 1. Classification scores for `model_3` test subsets

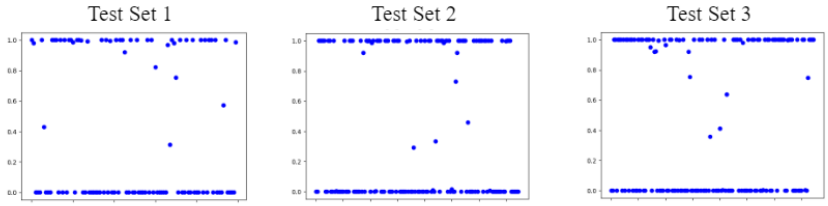


Figure 4. Classification for `model_3` test subsets

and recall measures. Classification obtained from `model_3` for each of three test sets is visualised in the figure 4.

4.3. Autoencoder in anomaly detection

As part of anomaly detection in weather data, two models were compared: for `model_1` time series was decomposed and proceeded further without a seasonal component, `model_2` included full weather data. The classical approach to the detection of abnormal observations assumes that such observations deviate from the mean by more than three standard deviation values. The `model_1` identified 14 observations, 13 of which exceeded the three sigma threshold for at least one feature. The `model_2` indicated 15 atypical observations, all of which accorded with the classical approach. However, these results cannot be considered positive, as the classical approach indicates the presence of 813 outlier observations. Since anomaly detection is based on the imperfect mapping of the input data, an attempt was made to deteriorate the model's performance by changing the learning parameters. Unfortunately, as a result of these changes, the model lost the ability to generalise and could not serve as a basis for further analysis.

5. Conclusions

An important part of building a model for data generated by a non-stationary process is to ensure robustness by choosing an appropriate loss function. The classification and associated automatic labelling of data produced positive results, it can be concluded that weather data are suitable for classification based on autoencoders. In the process of anomaly detection, the following problem arose: based on model optimized for input data reconstruction it was not possible to aptly detect the anomaly. This can be interpreted in two ways. Either the selected data are inadequate for autoencoder-based anomaly detection or the model distinguishes abnormal observations differently from classical statistical methods.

Acknowledgment

The author would like to thank dr Piotr Fulmański for his work as mentor of the project in Student Research Grants programme conducted by the University of Łódź.

References

[1] Polish institute of meteorology and water management. URL `https://danepubliczne.imgw.pl/data/dane_pomiarowo_obserwacyjne/dane_meteorologiczne/miesieczne/klimat/`.

[2] Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[3] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 833–840. Omnipress, Madison, WI, USA, 2011. ISBN 9781450306195.

[4] Huber, P. J. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi:10.1214/aoms/1177703732. URL `https://doi.org/10.1214/aoms/1177703732`.

Institutional Black Boxes Pose an Even Greater Risk than Algorithmic Ones in a Legal Context

Andrzej Porębski[0000−0003−0856−5500]

Jagiellonian University
Faculty of Law and Administration
Gołębia 24, 31-007 Kraków, Poland
and.porebski@uj.edu.pl

Abstract. *Black boxes in machine learning (ML) systems can be understood in at least two ways; in relation to (1) an algorithm, i.e., a decision rule, when that rule is impossible for a human to interpret, or (2) the secrecy of that rule (proprietary nature of it), due to business or economic factors. I call the first understanding "algorithmic black boxes" and the second "institutional black boxes". These two understandings are independent of each other, in particular, transparent algorithms can be part of systems that are institutional black boxes. I indicate that when it comes to the application of ML in public institutions applying the law (e.g., courts), institutional black boxes pose a particular threat to the integrity and reliability of ML systems used in such a context. I argue that in the eXplainable Artificial Intelligence trend, more attention should be paid not only to the favourable features of the algorithm (e.g., direct interpretability) but also to the business context in which the ML system is developed. Its secrecy can sabotage the transparency of even the simplest models.*
Keywords: *AI & Law, black box, transparency, explainability, COMPAS, eXplainable Artificial Intelligence, XAI*

1. Introduction

Today, probabilistic models created based on empirical data (machine learning models, MLM[1]) are starting to be used more widely in the legal context, and such

[1]In this paper, I am using the more precise concept of "machine learning model", instead of the vague notion of "artificial intelligence", which could sometimes be misunderstood [1], especially

DOI:10.17388/WUT.2024.0002.MiNI

use is increasingly being demanded [2, 3, 4]. As a result, the debate on the regulation of the use of MLM in supporting decisions made by judges and officials is becoming increasingly important. One of the key discussions about the regulation of such systems is the question of their fairness [5] and related features, among which it is worth highlighting transparency [4, 6].

In this paper, I would like to consider possible understandings of what a black box is, and then relate them to the use of MLM in a legal context. I use the term "institutional black box" to capture the numerous situations where technologies are even completely non-transparent due to business decisions rather than algorithmic characteristics. I call for a stronger emphasis on the institutional and business conditions for the development of ML-based systems to be used in such sensitive areas of life as law.

2. Key concepts

2.1. Transparency

The transparency of the system has been characterised in various ways in the literature, however, a crystal clear definition has not been achieved [7]. According to a meta-analysis by Arrieta et al. [8], a model is transparent when, without using additional techniques, it can be considered as understandable. As can be seen, this approach emphasises the characteristics of the MLM itself, without referring to organisational conditions. What is also a problem is that both "additional techniques" and "understandable" are vague concepts. Therefore, it may be a better idea to do one of two things: (1) define what non-transparent models are and then obtain transparent models by negation or (2) define transparent models by enumerating those classes that have favourable features from the point of view of interpretation and understanding.

If the first path is chosen, non-transparency can be considered to consist of excessive model complexity that makes the parameter space so expansive that it is impossible to interpret without the use of additional techniques. The first path has the potential advantage of allowing models that belong to classes that are traditionally considered transparent but are so much more extensive that they are difficult to interpret, to be included as non-transparent. For example, according to the first

by people without technical knowledge such as lawyers. However, the majority of the remarks presented in this text can also be applied to artificial intelligence systems since their base most often remains machine learning.

path, a decision tree with thousands of vertices will be non-transparent because it will be too complex, even though the exact decision function is known.

If one chooses the second path, i.e., to enumerate transparent models (inherently interpretable, fully interpretable), one can consult a fairly common consensus: several classes of models are typically included in this group [8, 9, 10]. Among them are linear and logistic regression, generalized additive models, and decision trees or k-nearest neighbours. Then, the other classes of models (e.g., artificial neural networks) are non-transparent.

However, the aforementioned ways of defining are technologically oriented and linked to the interpretability of the model. Meanwhile, we should keep in mind that the incomprehensibility of the system may also be due to the fact that the exact way the system works has been kept secret. Beaudouin et al. [11] define transparency in this context, indicating that transparency means that the operation of the algorithm and how it was created is "available for scrutiny". As they suggest, "transparency does not necessarily mean that the underlying information is easily comprehensible to humans" [11].

A better insight into two of the meanings of non-transparency will be provided by distinguishing between the two concepts of a black box.

2.2. Black box

Black boxes do not have a single consistent definition, but it is possible to distinguish two main understandings under which the term is used: (1) algorithmic, referring to the decision rules themselves, emphasizing the impossibility of human understanding of them (algorithmic black box), and (2) institutional, referring to such business conditions that make the rules secret (institutional black box). Black boxes are defined in a similar way, for example, by C. Rudin [6]. These two meanings are independent of each other.

It can be noted that non-transparent MLM will always be black boxes, but only in an algorithmic sense. Transparent MLM, on the other hand, will not be algorithmic black boxes but may remain institutional black boxes. The inability to trace the reasons why the system returned a certain result may, therefore, be due both to problems at the level of understanding of the effect of machine learning and also to the fact that this effect is of an undisclosed, proprietary nature.

2.3. Institutional black box

In the text, I would like to focus on institutional black boxes, which are less discussed within the XAI trend. Meanwhile, nowadays, due to the emergence of new tools, people are no longer powerless against the interpretation of algorithmic black boxes, as long as they have access to the exact form of the system. Algorithmic black boxes can be explained using post-hoc explaining methods [9, 10]. Moreover, if the data and the learning method are well-known, many of the system's actions can be indirectly interpreted on that basis. Finally, also in the case of algorithmic black boxes, it is possible to some extent to analyse for fairness such as the presence of algorithmic bias in the data [12]. Thus, it is not the case that classically defined black boxes (algorithmic ones) completely prevent insight into themselves, and in particular whether the system is likely to meet certain basic standards. However, in order to do so, a sufficiently large number of elements of the created system, as well as the method of its creation, must be public.

Meanwhile, in the case of institutional black boxes, the entity developing the system, most often citing business confidentiality, makes secret the detailed characteristics of the system as well as the process of developing it. However understandable and justifiable this may be from a business or economic perspective, it very significantly affects the explainability and understandability of the system. This creates a situation in which independent parties, and even the user, can learn only as much about the system and its decisions as the top-down system developer has planned. Significantly, an institutional black box will typically lead to the incomprehensibility of the system and its results even if the system is based on simple decision rules and fully directly interpretable functions (e.g., the logistic regression function from which a logistic classifier is derived).

3. The institutional black box in the legal context

The inadequacy of the algorithm-focused understanding of the black box is shown by the extensive discussion taking place (mainly in the USA) on whether non-transparent MLM can be used in risk assessment tools for the application of the law. This discussion revolves mainly around the COMPAS system, which is used in some courts in the US to predict the risk of recidivism (that is, which could be used by judges in making sentencing decisions) [4, 6, 13, 14]. On the one hand, as its developers argue, COMPAS is based on a simple, directly interpretable and traceable MLM (probably, it is a generalised additive model [13]), which is

"fully transparent to the agencies that use it" [15]. In this sense, they argue for the (algorithmic) transparency of the model. On the other hand, as many legal scholars argue, this algorithmic transparency is of no use when faced with the fact that neither the exact process of creating the system (including the data, its preprocessing, etc.) nor the final form of the model is actually known, because its full documentation is not disclosed [6, 13]. They, in turn, suggest that COMPAS is the (institutional) black box because it is proprietary. In this context, it is also very difficult to determine what the actual level of transparency is for the "agencies that use it", to which COMPAS creators refer.

This characterization of COMPAS creates very serious problems for the discussion of its fairness and, therefore, also for its legitimacy to support decisions relating to a critical spheres: the justice system and the freedom of the defendant. While it is theoretically possible to know the exact basis from which its prediction is derived, since it is algorithmically transparent, in practice these are implicit to the vast majority of people involved. It is not even entirely clear which actors have access to full knowledge of the system and to what extent. Faced with this situation, the public, interested in reliability, was condemned to only a few independently obtained data on the operation of the system and reconstructing its operation as if they were dealing with a black box further hidden in a dark room. Such a process of learning how the system works creates a special type of problem that would be unlikely to gain such scale if we were not dealing with an institutional black box. COMPAS was found to be racially biased in the course of the first investigation by journalists from the ProPublica service [16, 17]. Their work was cited thousands of times. Later, however, other studies rightly pointed out the methodological shortcomings of the ProPublica team's article, questioning the possibility of declaring it racially biased based on the analysis performed [6, 13, 18]. But since an institutional black box was involved in this situation, there was no easy way to straighten out this first, erroneous, information. The emerging declarations of the company developing the system [15, 19] may have had little credibility in a situation where it ultimately guarded access to the algorithm. At the same time, under conditions of emerging doubt, the most effective way to resolve it would have been to simply inform the public – which, under conditions of an institutional black box, was impossible. It is fortunate that the ProPublica researchers released a detailed description of their methodology and all the data used, making it transparent. Only by doing so was it possible to control its quality. This is a good illustration of the important role that institutional transparency has in issues related to data science in its broadest sense. If the methodology and

the data were not transparent in the case of the ProPublica analysis, it would be analogous to an "institutional black box" and involve all the related problems.

In the legal context, it is very unfortunate that the basis for such sensitive decisions is unclear. It is not just the lack of clarity in the algorithm itself but also that relating to the data used, how it was trained, and so on. So, it is even conceivable that for the legitimacy of the system and its fairness, an algorithmic black box, which is an institutional white box, would be better in such an application. This is because institutional transparency would allow for extensive public scrutiny and the use of explanatory tools such as post-hoc explainers. Institutional transparency will always provide a better chance of verifying the fairness of an algorithm (even a black box) than, e.g., journalistic investigations based on potentially incomplete data, and with methodological limitations. Hence, institutional transparency can be seen as a prerequisite for ML systems applied in a legal context.

4. Conclusions and practical implications

In the paper, I focused on the need to distinguish between two types of "black box". The first, most commonly used in the literature, refers to the characteristics of the technology (algorithmic black box), and the second to the characteristics of the organizational processes that make the ML system proprietary (institutional black box), even if it could be fully interpreted after the relevant data on the process of its construction and the decision rules created were revealed.

I have argued that institutional black boxes are potentially even more threatening to the transparency and ethics of emerging systems than the complexity of the algorithm itself. In some cases, institutional rather than algorithmic black boxes are the reason why ML systems operating in sensitive contexts such as law are non-transparent, and thus widely criticized. Therefore, in the discussion within the eXplainable Artificial Intelligence trend (XAI) [20] more attention should be paid to the institutional considerations for developing critical decision support systems. As Longo et al. [20] correctly indicate, one of the key problems for the XAI is "addressing the power imbalance between individuals and companies". Institutional black boxes seem to be one way in which such imbalances arise.

Decision makers, when designing regulations for the use of decision support systems in the operations of the public administration or the judiciary, should take into account the need to ensure that the process of developing such solutions is open and transparent. Thanks to this transparency and the open discussion of ML

based decision support systems, they will be able to gain proper legitimacy even if their basis is an algorithmic black box, and, above all, they will be able to be traced to the correctness of their operation and the fairness of the decisions. In order to achieve these values, it is worth considering the assistance of academia as well as the self-development of systems by the government, thereby circumventing business secrecy issues.

Acknowledgment

This research was funded by the National Science Center, Poland, and is the result of the research project "The Understandability Requirement of Machine Learning Systems Used in the Application of Law" (no. 2022/45/N/HS5/00871).

References

[1] Bewersdorff, A., Zhai, X., Roberts, J., and Nerdel, C. Myths, mis- and preconceptions of artificial intelligence: A review of the literature. *Computers and Education: Artificial Intelligence*, 4:100143, 2023. doi:10.1016/j.caeai.2023.100143.

[2] Berk, R. and Hyatt, J. Machine Learning Forecasts of Risk to Inform Sentencing Decisions. *Federal Sentencing Reporter*, 27(4):222–228, 2015. doi:10.1525/fsr.2015.27.4.222.

[3] Cohen, M. C., Dahan, S., Khern-Am-Nuai, W., Shimao, H., and Touboul, J. The use of AI in legal systems: determining independent contractor vs. employee status. *Artificial Intelligence and Law*, 2023. doi:10.1007/s10506-023-09353-y.

[4] Porębski, A. Machine learning and law. In *Research Handbook of Law and Technology*, pages 450–467. Edward Elgar Publishing Cheltenham, 2023. doi:10.4337/9781803921327.00037.

[5] Berk, R., Heidari, H., Jabbari, S., Kearns, M., and Roth, A. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research*, 50(1):3–44, 2021. doi:10.1177/0049124118782533.

[6] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215, 2019. doi:10.1038/s42256-019-0048-x.

[7] Bellucci, M., Delestre, N., Malandain, N., and Zanni-Merk, C. Towards a terminology for a fully contextualized XAI. *Procedia Computer Science*, 192:241–250, 2021. doi:10.1016/j.procs.2021.08.025.

[8] Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, 58, 82–115, 2020. doi:10.1016/j.inffus.2019.12.012.

[9] Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., and Hussain, A. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation*, 16(1):45–74, 2024. doi:10.1007/s12559-023-10179-8.

[10] Maksymiuk, S., Gosiewska, A., and Biecek, P. Landscape of R packages for eXplainable Artificial Intelligence. *arXiv preprint*, 2009.13248, 2020. doi:10.48550/arXiv.2009.13248.

[11] Beaudouin, V., Bloch, I., Bounie, D., Clémençon, S., d'Alché-Buc, F., Eagan, J., Maxwell, W., Mozharovskyi, P., and Parekh, J. Flexible and Context-Specific AI Explainability: A Multidisciplinary Approach. *arXiv preprint*, 2003.07703, 2020. doi:arXiv.2003.07703.

[12] Aggarwal, A., Lohia, P., Nagar, S., Dey, K., and Saha, D. Black box fairness testing of machine learning models. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 625–635. Association for Computing Machinery, 2019. doi:10.1145/3338906.3338937.

[13] Rudin, C., Wang, C., and Coker, B. The Age of Secrecy and Unfairness in Recidivism Prediction. *Harvard Data Science Review*, 2(1), 2020. doi:10.1162/99608f92.6ed64b30.

[14] Rudin, C., Wang, C., and Coker, B. Broader Issues Surrounding Model Transparency in Criminal Justice Risk Scoring. *Harvard Data Science Review*, 2(1), 2020. doi:10.1162/99608f92.038c43fe.

[15] Jackson, E. and Mendoza, C. Setting the Record Straight: What the COMPAS Core Risk and Need Assessment Is and Is Not. *Harvard Data Science Review*, 2(1), 2020. doi:10.1162/99608f92.1b3dadaa.

[16] Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine Bias. *ProPublica*, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[17] Larson, J., Mattu, S., Kirchner, L., Angwin, J. How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*, 2016. URL https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[18] Flores, A.W., Bechtel, K., and Lowenkamp, C.T. False Positives, False Negatives, and False Analyses: A Rejoinder to "Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks". *Federal Probation*, 80(2):38–46, 2016. URL https://www.uscourts.gov/sites/default/files/80_2_6_0.pdf.

[19] Dietrich, W., Mendoza, C, and Brennan, T. COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity. *Northpointe*, 2016. URL https://www.equivant.com/response-to-propublica-demonstrating-accuracy-equity-and-predictive-parity.

[20] Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Del Ser, J., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., and Stumpf, S. Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106:102301. 2024. doi:10.1016/j.inffus.2024.102301.

Author Index

ISBN 978-83-8156-697-1