Konferencijos

„Lietuvos magistrantų informatikos ir IT tyrimai“ darbai

2024 m.

VILNIAUS UNIVERSITETO LEIDYKLA

eISSN 2783-784X

Konferencijos **„Lietuvos magistrantų informatikos ir IT tyrimai“** darbai

2024 m. gegužės 10 d.

VILNIAUS
UNIVERSITETO
LEIDYKLA

2024

eISSN 2783-784X
https://doi.org/10.15388/LMITT.2024

Įžanga

Konferencija „Lietuvos magistrantų informatikos ir IT tyrimai“ – tai jau penktą kartą vykstantis renginys, skirtas pristatyti naujausius studentų pasiekimus informatikos ir IT srityse. Šio renginio tikslas – ugdyti studentų mokslinių darbų rengimo bei rezultatų pristatymo įgūdžius, supažindinti su kitų studentų atliekamais tyrimais, paskatinti studentus domėtis moksline veikla. Renginys subūrė studentus iš kelių Lietuvos universitetų. Konferencijoje savo pranešimus skaitė studentai iš Vytauto Didžiojo universiteto, Vilniaus Gedimino technikos universiteto ir Vilniaus universiteto. Konferencijoje aptariamos įvairiausios temos, nuo programų sistemų kūrimo iki dirbtinio intelekto, ir tai suteikia puikią galimybę dalyviams pagilinti savo žinias, keistis idėjomis bei užmegzti vertingus profesinius ryšius.

Konferenciją organizuoja Lietuvos mokslų akademija (LMA) kartu su Vilniaus universitetu. LMA – tai įstaiga, jungianti žymiausius Lietuvos ir savo veikla su Lietuva susijusius mokslininkus. Ji yra nepriklausoma Lietuvos Respublikos Seimo, Vyriausybės ir jai pavaldžių institucijų ekspertė ir patarėja mokslo bei studijų, kultūros, socialinės raidos, ūkio, gamtosaugos, sveikatos apsaugos, technologijų bei kitais klausimais. LMA įgyvendina šalies mokslui ir eksperimentinei plėtrai reikšmingus ES struktūrinių fondų projektus, rengia šalies bei tarptautines mokslines konferencijas, užsienio mokslininkų seminarus, mokslininkų susitikimus, akademinius skaitymus, parodas. Tikime, kad dalyvavimas šioje LMA kuruojamoje konferencijoje paskatins magistrantus ir kitus studentus tęsti mokslinę veiklą ir pabaigus studijas.

Konferencijos darbuose publikuoti recenzuoti studentų parengti moksliniai straipsniai. Tai dažniausiai pirmosios mokslinės publikacijos, bet tikimės, kad ateityje virs į straipsnius prestižiniuose mokslo žurnaluose. To norėtume palinkėti konferencijos dalyviams.

Organizatoriai

dr. Jolita Bernatavičienė
prof. habil. dr. Gintautas Dzemyda
prof. dr. Olga Kurasova
prof. dr. Julius Žilinskas

Contents

Method for Determining the Level of Centralization in BTC Lightning Nodes: A Centrality Analysis of the Lightning Network

Laura Atmanavičiūtė, Tomas Vanagas, Saulius Masteika

Vilnius University, Kaunas Faculty, Institute of Social Sciences and Applied Informatics
Muitinės str. 8, LT-44280 Kaunas, Lithuania
laura.atmanaviciute@knf.vu.lt, tomas.vanagas@knf.vu.lt, saulius.masteika@knf.vu.lt

Abstract. This study explores the Bitcoin Lightning Network (BLN), a Layer 2 solution for faster and cheaper transactions. Concerns about centralization have emerged due to the increasing concentration of power among specific nodes, named "hubs." Statistical measures like the Gini coefficient reveal a trend towards centralization, challenging the LN's decentralized nature. Consequently, further research is necessary to address this issue and ensure the integrity of the LN architecture. This paper aims to establish a method for determining the level of centralization within the BLN by applying centrality analysis techniques. Study revealed that over the six-year period Gini coefficient increased from 0.87 to 0.955, indicating significant inequality and apparent centralization of BLN nodes.

Keywords: Bitcoin, Lightning Network, Centralization, Data processing

1 Introduction

The Bitcoin Lightning Network (BLN) is a promising Layer 2 solution built on Bitcoin (BTC) that aims to make transactions faster and cheaper. When it was first introduced, LN fees were expected to be much lower than standard BTC transactions [1]. It was meant to be a way for users to send money to each other directly, without loading the transaction data to the whole BTC network [2]. Two users can agree to establish a direct channel by creating a multi-signed transaction on the blockchain [3]. When the channel is closed, only the final balance needs to be settled on-chain as a single transaction. Consequently, the system becomes capable of accommodating a significantly greater volume of payments [4].

DOI: https://doi.org/10.15388/LMITT.2024.1

However, a potential centralization issue has emerged. There is no robust answer as to whether the current distribution within the LN indicates a trend towards centralization, with a select few nodes maintaining a disproportionate share of the network's total channel capacity [5]. Can powerful, well-funded nodes, acting as hubs with extensive payment channels that process a large volume of transactions, gain undue influence? This dominance by a few hubs might lead to a more centralized system, contradicting the decentralized nature of BTC itself [6]. Based on this, the following hypothesis is proposed:

Hypothesis. An unequal distribution of channels among nodes within the BLN may be a contributing factor to a decrease in the network's overall decentralization.

To address the question of centralization, it's important to delve into the methods and ways, using specific coefficients, to measure the level of centralization. The challenge here is significant, as the data must first be extracted from the blockchain, categorized, and linked to obtain variables that can be appropriately integrated into methodology.

The aim of this study is to establish a method for determining the level of centralization within the BLN by applying centrality analysis techniques. Critical tasks towards achieving this goal include:

- Developing a comprehensive method for calculating centralization.
- Extracting, gathering, linking, and storing data from the BTC blockchain Layer 1 (L1) and Layer 2 (L2).
- Conducting experimental calculations and providing visual representation of the results.

This paper proposes a method for determining the centralization level of BLN nodes. It combines Gini coefficient to quantify the centralization and the Lorenz curve to visually present the results. The structure of the paper is as follows: The first part of the paper introduces the topic, outlining the research focus. The second part delves into the background of the method, analysing centrality aspects and coefficients relevant to assessing the BLN's centralization level. The third part consists of detailed explanation of data extraction and linking, including a proposed data retrieval and storage scheme. The fourth section presents the experimental setup and its results of the experiment. Finally, the last section presents the conclusions of the study.

2 Background of the method

To objectively assess centralization within the LN, it's important to explore different aspects of centrality. There are five main aspects that can be considered when assessing the level of centralization of the BLN – degree, weighted degree, betweenness, eigenvector and closeness centrality. **Degree centrality** evaluates the number of channels a node has with other nodes – it identifies highly connected nodes but not the significance of those connections [2]. This limitation can be addressed by considering **weighted degree centrality**, which incorporates channel capacity into calculations [7]. **Eigenvector centrality** is variant of degree centrality and measures a node's influence in the network based. In simpler terms, degree centrality counts nodes and eigenvector centrality measures the influence of a node [8]. Meanwhile, **closeness** and **betweenness centrality** consider the distance between nodes when trying to find the shortest connection between them [7]. **Closeness centrality** is used for calculating how close a node is to all other nodes in the network [2]. It helps understand the efficiency of network, but it might not directly address the concern of the centralization if all nodes have similar closeness. Another approach is **betweenness centrality**, which measures how frequently a node is on the shortest path connecting other nodes – in the context of the LN, it indicates a node's significance for routing payment [9, 10]. For this paper, specifically weighted degree centrality aspect is employed, because it counts not only the number of channels a node has, but also considers the capacity of each connection.

After choosing to assess BLN centralization through weighted degree centrality aspect, it is important to delve into coefficients which can be employed to quantify this centrality measure. One of the most common coefficients is Gini coefficient, which measures the inequality of channel distribution within the LN. Higher Gini coefficient suggests a greater centralization – it is known as a strong indicator of overall network centralization, especially if utilized with other measures [2, 11]. Another well-known coefficient is Herfindahl–Hirschman index (HHI), which can also be used as a method when determining network's centralization. HHI is a traditional metric used to assess market's concentration and is often used to measure market efficiency [12].

Gini coefficient is more insightful than the HHI as it directly measures inequality in capacity distribution and shows how influential few nodes in the network might be. Meanwhile, HHI is less sensitive to imbalances, which is one

of the main issues of network centralization. Additionally, the Gini coefficient acts like a score between 0 and 1, where 0 signifies everyone having an equal share of the resource and 1 represents a scenario where one individual has everything [2, 9], which makes this measure easy to interpret, while HHI doesn't have a straightforward interpretation. A lower Gini coefficient points towards a network with a more balanced distribution (decentralized), while a higher value suggests an uneven spread (centralized) of the resource being analysed. It can be measured using the following formula:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2N^2 \bar{x}}$$

N is used to represent a total number of nodes, x_i and x_j represent capacity of nodes and $\bar{x}$ is an average capacity across all nodes.

The Gini coefficient represents the difference between the line representing perfect equality and the actual distribution depicted by the Lorenz curve. It's calculated by subtracting the area below the Lorenz curve from the area below the line of perfect equality, and then dividing this result by the total area under the line of perfect equality [10]. The Lorenz curve visually illustrates how channels are distributed among nodes based on the size of their channel capacity. It compares this distribution to a perfectly equal scenario represented by a line at a 45-degree angle, known as the line of equality. The area between this line and the Lorenz curve is utilized to calculate the Gini index [13]. The analysis of the Gini coefficient and Lorenz curve for channel capacity distribution is leveraged to determine the level of centralization of BTC lightning nodes.

Existing research [2, 6, 9, 10, 11] utilizes the Gini coefficient revealing a growing trend of uneven distribution of channel capacity within the LN. While these studies provide valuable insights, there still are some limitations – such as detailed descriptions of data processing, which limits replication of study. This paper addresses these gaps by proposing a data collection and linking scheme, along with different timestamps than compared to related research.

3 Data processing for proposed method

To research the centralization level of the BLN, data is gathered from 3 primary sources – LN Research [14], Bitcoin Core [15] and Electrum Node [16]. LN Research investigates the LN data, while Bitcoin Core validates transactions and confirms blocks. Electrum Nodes act as intermediaries –

they don't store entire blockchain and are relevant for this paper to collect spending transaction information. Data retrieval and storage is presented in Figure 1.

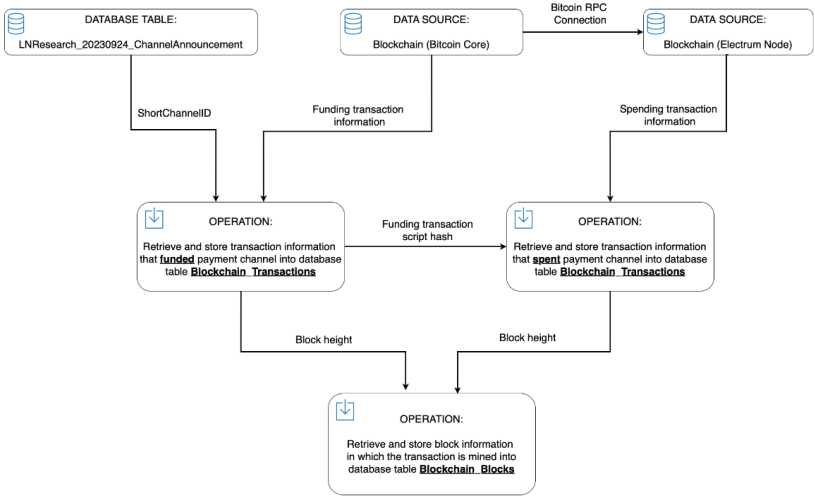


Figure 1. Data retrieval and storage

First, L2 data from LN Research is gathered. This research provides data of LN transactions by exchanging information through gossip protocol. For this study, relevant are 'Channel Announcement' messages. They provide information about the creation of a new payment channel in the LN – such as unique identifier (ID) of the channel ('ShortChannelID'), and nodes, which participates in this channel IDs.

While LN Research provides all relevant information about LN transactions, it lacks data about the L1 in which these transactions take part. To collect relevant data from L1, operating system 'MyNodeBTC' is utilized which has Bitcoin Core and Electrum Server installed. It helps to synchronize a Bitcoin's full node and an Electrum node for the transaction indexing. The BTC blockchain keeps a transaction dataset, which includes all transactions that have transpired on the BTC network. This database records all critical data of each transaction, including **timestamps** – the precise date and time at which BTC was locked within a transaction, **transaction amounts** – the quantity of BTC that was locked, and **channel status** – an indicator which indicated whether the transaction was utilized for opening a LN channel

(spent) or is still unused (unspent). Transactions which were identified as spent, were further investigated by assigning the specific block height within the blockchain, where the spending transaction has occurred.

Bitcoin Core software was configured to index all transactions in the blockchain by enabling 'txindex' flag in the configuration, while Electrum Server has various indexes to support Electrum Light Wallets, including transaction script hash index. Using Bitcoin Core and Electrum Server node it is possible to efficiently retrieve BLN payment channel funding and spending transaction data from the blockchain without fully scanning entire blockchain, as Electrum Server can leverage its prebuilt indexes.

3.1. Data linking

The data is linked by connecting data collected by LN Research to the relevant blockchain transactions which opened the channels. Blockchain data retrieval process starts by iterating through every 'Channel Announcement' message in the LN research dataset and retrieving transaction which opened BLN payment channel from Bitcoin Core node. The link is facilitated by the 'ShortChannelID', which consists of the block height, the transaction index within the block, and the transaction output index.

Bitcoin Core's node is requested to return transaction based on block height, transaction index in the block and transaction output index in the transaction ('ShortChannelID') and inserting returned data to the database table 'Blockchain_Transactions'. After inserting the channel funding transaction details, it is required to find when transaction output in question was spent. For this part of the process, Electrum Server Node can return this data by using RPC's 'blockchain.scripthash.get_history' function.

At the end of the blockchain data retrieval process there should be same number of records in both 'LNResearch_20230924_ChannelAnnouncement' and 'Blockchain_Transactions' database tables. This verifies that data from both sources has been successfully linked.

After the transaction details have been retrieved, it is necessary to retrieve data about the block in which the transaction has occurred. Information about the blockchain block contains a timestamp which shows when the transactions in question have been mined. Data about the blockchain blocks are stored in a different database table 'Blockchain_Blocks'.

The full dataset of both L1 and L2 allows researchers to have a full picture of the BLN by joining database tables together and filtering the dataset to any moment of time of BLN existence and applying calculations.

4 Experimental setup and results

To capture the changes of the LN, six data snapshots were selected for the experiment, taken on June 1st of each year, starting from 2018 – the year when LN was presented – and ending with 2023. This approach not only allows tracking changes and identifying trends in the LN structure over time, but it also addresses the challenge of the LN's rapidly evolving data. The experiment analyses the distribution of channel capacity across the BLN nodes. The experiment utilizes a dataset containing 495,755 channel announcement records. A node is considered existing if it has at least one channel open during that timeframe.

The results reveal a concerning trend towards centralization. Figure 2 presents Lorenz curves for the BLN nodes on weighted degree centrality aspect captured at six specific timestamps. It was created by retrieving data from the intermediate database table at specific moments of time. After this, all the nodes were sorted in ascending order based on the BTC amount and then cumulative percentages of the whole network were calculated in 1% granularity to calculate Lorenz curve. Figure 2 shows that over the time Lorenz curve is progressively moving further away from the perfect equality line across the six timestamps, which means that inequality between BLN nodes is increasing. This trend also aligns with the results of calculated Gini coefficient, which increased from 0.87 in 2018 to 0.955 in 2023, with an average of 0.926. This indicates a great inequality, suggesting a concentration of channel capacity among specific groups of nodes.

Furthermore, Figure 3 visually represents the results of Gini coefficients of weighted degree centrality for the BLN nodes and reinforces the observation. In the two years of LN, inequality for BLN nodes increased significantly. Starting with 0.87 Gini coefficient at the start of LN and reaching 0.936 in 2020, there is a huge decrease in network's decentralization.

The experimental results proves that the BLN is exhibiting tendencies towards the centralization, especially in channel capacity distribution. The research confirms the initial hypothesis – the Gini coefficient rose significantly over time and indicated unequal distribution of channel capacity among nodes. This distribution, which also was visualized by Lorenz curves deviating further from perfect equality, aligns with the hypothesis that uneven channel distribution is a contributing factor to a decrease in the network's overall decentralization.

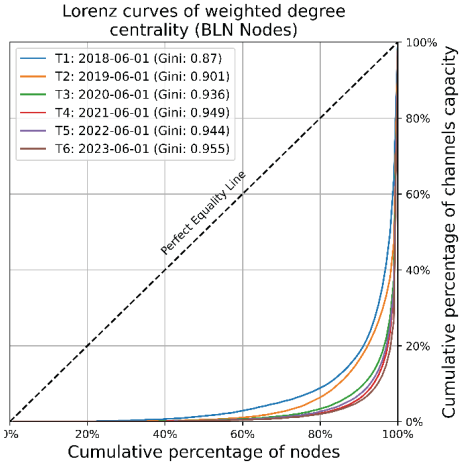


Figure 2. Lorenz curves of weighted degree centrality for Bitcoin Lightning Network nodes

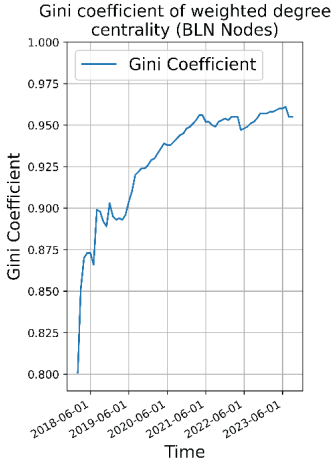


Figure 3. Gini coefficient of weighted degree centrality for Bitcoin Lightning Network nodes

5 Conclusions

This study established a method for assessing the level of centralization in BTC lightning nodes using the Gini coefficient and Lorenz curve. Building upon prior research that identified a concerning trend of uneven channel capacity distribution, this paper expands that research with a detailed data collection and linking scheme, along with a different range of timestamps. Gini coefficient was proposed as a reliable method because it measures inequality and shows how influential nodes in the network can be. Along with the Gini coefficient, the Lorenz curve depicted channel capacity distribution. This combined approach enabled comprehensive analysis and trend identification.

Data from both the BTC blockchain L1 and L2 was extracted, gathered, linked, and stored successfully by connecting LN Research data and queries to Bitcoin Core and Electrum Server nodes. This dataset ensured a complete picture of the BLN for further calculations assessing the centralization of the network.

The experimental calculations, using the Gini coefficient and Lorenz curves for the six timestamps, confirmed the initial hypothesis. The research revealed an increase in the Gini coefficient – from 0.87 in 2018 to 0.955 in 2023, signifying a growing inequality in channel capacity distribution among

nodes. This trend is also highlighted by Lorenz curves, which is progressively moving further away from perfect equality. The results of the experiments suggest a concerning shift towards centralization within the BLN – especially regarding channel capacity distribution among nodes.

References

[1] Poon J, Dryja T. (2016). The Bitcoin Lightning Network: Scalable Off-Chain. DRAFT Version 0.5.9.2. 2016. URL: https://lightning.network/lightning-network-paper.pdf.

[2] Lin, J.-H., Primicerio, K., Squartini, T., Decker, C., & Tessone, C. J. (2020). Lightning network: a second path towards centralisation of the Bitcoin economy. New Journal of Physics, 22(8), 83022. doi:10.1088/1367-2630/aba062

[3] Martinazzi, S., Flori, A. (2020). The evolving topology of the Lightning Network: Centralization, efficiency, robustness, synchronization, and anonymity. PloS One, 15(1), e0225966–e0225966. doi:10.1371/journal.pone.0225966

[4] Divakaruni, A., Zimmerman, P. (2022). The Lightning Network: Turning Bitcoin into money. Finance Research Letters 52.

[5] Carotti, A., Sguanci, C., & Sidiropoulos, A. (2023). Rational Economic Behaviours in the Bitcoin Lightning Network. doi:10.48550/arxiv.2312.16496

[6] Masteika, S., Rebždys, E., Driaunys, K., Šapkauskienė, A., Mačerinskienė, A., & Krampas, E. (2023). Bitcoin double-spending risk and countermeasures at physical retail locations. International Journal of Information Management, 102727. doi:10.1016/j.ijinfomgt.2023.102727

[7] Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245–251. doi:10.1016/j.socnet.2010.03.006

[8] Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. The Journal of Mathematical Sociology, 2(1), 113–120. doi:10.1080/0022250X.1972.9989806

[9] Mahdizadeh, M. S., Bahrak, B., & Sayad Haghighi, M. (2023). Decentralizing the lightning network: a score-based recommendation strategy for the autopilot system. Applied Network Science, 8(1), 73–33. doi:10.1007/s41109-023-00602-2

[10] Zabka, P., Foerster, K.-T., Decker, C., & Schmid, S. (2022). Short Paper: A Centrality Analysis of the Lightning Network. In Financial Cryptography and Data Security (pp. 374–385). Cham: Springer International Publishing. doi:10.1007/978-3-031-18283-9_18

[11] Lin, J.-H., Marchese, E., Tessone, C. J., & Squartini, T. (2022). The weighted Bitcoin Lightning Network. Chaos, Solitons and Fractals, 164, 112620. https://doi.org/10.1016/j.chaos.2022.112620

[12] Cheng, L., Zhu, F., Liu, H., & Miao, C. (2021). On Decentralization of Bitcoin: An Asset Perspective. doi:10.48550/arxiv.2105.07646

[13] Juodis, M., Filatovas, E., & Paulavičius, R. (2024). Overview and empirical analysis of wealth decentralization in blockchain networks. ICT Express, 00(00), 1–7. doi:10.1016/j.icte.2024.02.002

[14] Decker, C. (2023). Lightning Network Gossip. URL: https://github.com/lnresearch/topology

[15] Bitcoin Core Developers. (2024). Bitcoin Core. URL: https://bitcoincore.org/

[16] T. Voegtlin. (2011). Electrum Bitcoin Wallet. URL: https:// electrum.org/

Machine Learning Approaches in Atrial Fibrillation Detection

Deividas Butkus, Jolita Bernatavičienė

Vilnius University, Institute of Data Science and Digital Technologies
Akademijos str. 4, Vilnius
deividas.butkus@mif.stud.vu.lt

Abstract. Atrial fibrillation (AF) characterized by rapid and irregular electrical activity in the atria represents a prevalent form of cardiac arrhythmia that significantly challenges healthcare systems due to its links to heightened mortality and morbidity rates. Early detection of AF is critical for accurate and effective management and treatment. In response to this pressing need, numerous researchers have used machine learning (ML) to enhance the precision and efficiency of AF detection. By analyzing available datasets, signal lengths, preprocessing techniques, and a diverse array of ML approaches, this paper aims to cover methodologies of AF detection using electrocardiogram (ECG) data and ML.

Keywords: atrial fibrillation, machine learning, ECG, artificial intelligence, deep learning

1 Introduction

Atrial fibrillation (AF) is the most common cardiac arrhythmia seen in clinical settings leading to serious health problems such as stroke, heart failure, and increased mortality rates. The electrocardiogram (ECG) provides a graphical representation of the heart's electrical activity over time and is the standard diagnostic tool for detecting AF. The normal electrocardiogram in sinus rhythm depicted in Figure 1 comprises a P wave, a QRS complex, and a T wave. The QRS complex is often, but not always, three separate waves: the Q wave, the R wave, and the S wave. When AF is present, ECG usually comprises the absence of a P wave (while the QRS complex and T waves remain present), and an irregular pattern of R-waves.

The diagnosis of AF based on ECG signals requires the expertise of a trained specialist, typically a doctor, making it a time and resource-intensive process. The need for human interpretation poses challenges, including potential delays in diagnosis and limitations in scalability. Given these

DOI: https://doi.org/10.15388/LMITT.2024.2

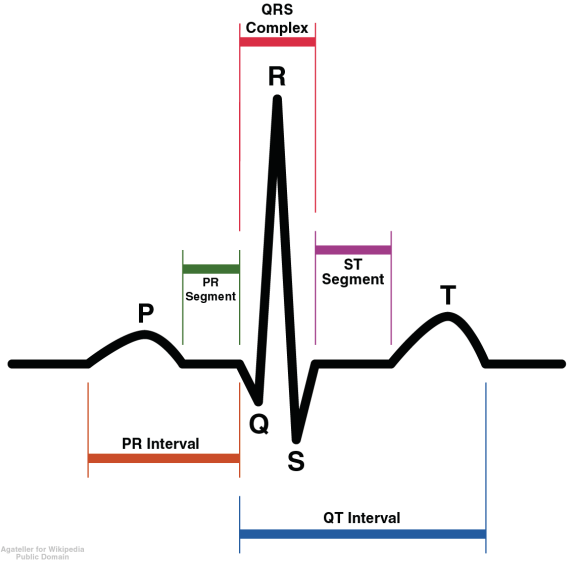


Figure 1. Schematic diagram of normal sinus rhythm for a human heart as seen on ECG [1]

challenges and the evolving landscape of healthcare technology, there is an increasing recognition of the potential role of machine learning algorithms in AF detection. Understanding nuances of the development of machine learning (ML) algorithms capable of identifying AF from ECG data with high accuracy is crucial for the advancement of automated diagnostic tools. This paper explores various machine learning approaches applied to ECG data for the detection of AF, aiming to systematically present advancements in AF detection in the ML field. Through an examination of different methodologies, from data preprocessing to model selection, this analysis seeks to review the most common strategies for this vital diagnostic task.

2 Literature review

2.1 Databases

Many recent publications in the field of ML for AF detection use public datasets provided by PhysioNet [2], a data repository for biomedical research. Based on the research by [3], the MIT-BIH Atrial Fibrillation Database (AFDB) [4] and the Computing in Cardiology (CinC) Challenge 2017 Database [5] are the two

most popular databases for AF detection, used in around 50 studies out of 147 reviewed, and both provided by PhysioNet. The AFDB database consists of 25 long-term ECG recordings of human subjects with AF, while the CinC Challenge 2017 contains a training set with 8528 single lead ECG recordings (normal (5076), AF (758), other (2415), and noisy (279)) lasting 9–60 s and a test set with 3658 ECG recordings of similar lengths that have been retained as a hidden test set. This training dataset is unbalanced and skewed towards the normal sinus rhythm class. Other databases, such as MIT-BIH Arrhythmia (MITDB) [6], and MIT-BIH Normal Sinus Rhythm (NSRDB) [7] are also often used. Based on the recent review of ML in AF detection, 10 out of 14 papers reviewed also used the AFDB Database, showing the continued relevance and popularity of this database in the field [8].

2.2 Signal length

The determination of an ECG signal length for the detection of AF using ML can depend on several factors, such as the objectives of the study, and the practical considerations of the machine learning algorithm. The authors of [9] tested various signal lengths for paroxysmal atrial fibrillation classification and got the best results with a 4 s window using the Second-Order System (SOS) algorithm. Another study by [10] employed both 2 s and 5 s windows using Convolutional Neural Network (CNN) and found that the 2 s segments achieved a higher specificity while 5 s segments showed a slightly better overall accuracy and sensitivity. However, study by [11] also used CNN and varying windows of 9–60 seconds and concluded that it was difficult to distinguish AF from other rhythms on small signal segments. Longer signal of 31 heartbeats was also backed up by [12] where the combination of CNN and Recurrent-Neural Networks (RNN) achieved specificity and sensitivity of 98.96% and 86.04% on unseen data. This variation in segment window and ML methods underscores the absence of a universal standard in the selection process. Optimal signal duration should satisfy the prerequisites of machine learning algorithms while simultaneously capture the dynamic nature of AF to facilitate precise classification.

2.3 ECG preprossessing techniques and features extraction

Morphological characteristics of the ECG are often derived and widely used in ML-based AF detection. One part of them is called time-domain transformations and includes RR interval, Heart Rate Variability (HRV), and

P-wave. Another group of transformations work on the frequency domain to detect high vs. low-frequency segments of the ECG and requires the use of Fourier Transform (FT) or Wavelet Transform (WT). These two domains, separately or together, are used widely in the research [13], [14, 15], [16]. In recent years, models such as CNN and RNN have been employed that directly process raw ECG signals, simplifying the detection process by eliminating the need for complex preprocessing steps [17], [18], [19]. This shift from complex preprocessing techniques to methods requiring minimal or no ECG preparation highlights an increasing trend in developing more efficient and accurate machine learning-based approaches for AF detection.

2.4 Machine learning algorithms

Machine learning algorithms for AF detection have demonstrated significant diversity and innovation, integrating traditional machine learning approaches with sophisticated deep learning models to enhance diagnostic accuracy. Deep learning methods, particularly CNN and Long Short-Term Memory (LSTM) networks, have shown to surpass traditional classifiers like Multilayer Perceptrons (MLP) and logistic regression in effectively processing ECG signals for AF detection, indicating a shift towards more complex models for better performance [20].

Further advancements include the use of CNNs in both single-channel and innovative two-channel models. A two-channel CNN model, for instance, uses one channel to identify where to look for the detection of AF in the ECG, while the other performs the actual detection [19]. Additionally, CNNs have been employed directly on raw ECG waveforms, bypassing traditional feature extraction processes [21], and in combination with RNN for extracting high-level features from RR intervals [12]. The study by [17] combined CNN and bagged tree ensemble to classify a filtered ECG signal - if the confidence of the classifier reaches a certain threshold, CNN is used, otherwise 43 PQRS features are used in a bagged tree ensemble.

Ensemble models and decision trees have also been instrumental in AF detection, utilizing a combination of expert features, signal processing methods, and learned features. These models benefit from the ensemble strategy by integrating multiple classifiers to improve prediction outcomes, evidenced by the use of bagged tree ensembles, gradient-boosted tree, and random forest classifiers based on hand-crafted and selected features for reliable AF detection [22], [23], [24].

Recent developments in the field have introduced innovative methods to enhance the detection and classification of AF. One such method utilizes a deep residualdense network based on a bidirectional recurrent neural network (Bi-RNN). This approach combines one-dimensional dense residual networks with Bi-RNNs and attention mechanisms to enable end-to-end feature learning from ECG signals, simplifying labour-intensive feature extraction steps [25]. Another approach merges the strengths of multilayer CNNs and RNNs with LSTM capabilities into a singular classifier. First, the multilayer CNN is utilized for extracting high-level features from the raw input sequence and then the RNN structure known as LSTM is used for processing the sequential features extracted by CNN. Lastly, the logistic classifier provides the posterior probability of the input sequence containing AF with notable sensitivity, specificity, and accuracy rates [26].

In conclusion, the automated AF detection landscape shows significant variation across ML algorithms without a single best method for AF detection. This diversity is highlighted in Table 1, which collates and compares different databases, signal lengths, preprocessing techniques, and machine learning algorithms with their corresponding performance metrics.

2.5 Evaluation metrics

The evaluation of ML algorithms in the context of AF detection is crucial to ensure their reliability and effectiveness in clinical applications. Traditional metrics for assessing the accuracy of detection methods in medicine include sensitivity, specificity, positive predictive value, and accuracy [29], [30]. These metrics serve to gauge:

Table 1. Summary of AF detection studies with percentage evaluation scores.

Author	Dataset	Signal length	Features	ML algorithm	Eval. metric	Score
Andersen et al (2017) [15]	AFDB	60, 100, and 300 beats	Time-domain, frequency-domain features	Support Vector Machine	Se, Sp	96.81%, 96.20%
Zabihi et al (2017) [24]	CinC Challenge 2017	5s segments with 4s overlap	Time-domain, frequency-domain, nonlinear features, meta-level features, morphological features	Random forest	F1	84%

Author	Dataset	Signal length	Features	ML algorithm	Eval. metric	Score
Hong et al (2017) [22]	CinC Challenge 2017	20 s	Expert features, center-wave features, and DNN features	Decision trees	F1	85%
Kamaleswaran et al (2018) [21]	CinC Challenge 2017	9s to 61s	Raw ECG data at various sampling frequencies	CNN	F1	82%
Kropf et al (2018) [23]	CinC Challenge 2017	Not specified	time-domain, frequency-domain, and morphological features	Gradient boosted tree	F1	84%
Plesinger et al (2018) [17]	CinC Challenge 2017	6 s	PQRS features or a 9-times filtered ECG signal	Bagged tree or CNN	F1	82%
Andersen et al (2017) [15]	AFDB	60, 100, and 300 beats	Time-domain, frequency-domain features	Support Vector Machine	Se, Sp	96.81%, 96.20%
Mousavi et al (2019) [19]	AFDB	5 s	Raw ECG data	ECGNET	Se, Sp, Acc	99.53%, 99.26%, 99.40%
Laghari et al. (2023) [25]	CPSC2018 [27]	Not specified	Residual dense CNN and RNN features	Residual dense CNN and RNN	Se, Sp, Acc	93.09%, 98.71%, 97.72%
Kumar et al. (2023) [26]	CACHET-CADB [28], AFDB, NSRDB, MITDB	Interval of 30 RR	Multilayer CNN features	CNN and RNN	Se, Sp, Acc	96.06%, 98.29%, 97.04%

- Sensitivity (Se), also known as true positive rate (TPR) or recall, measures the proportion of signals correctly classified as AF versus the actual number of signals identified as AF.
- Specificity (Sp) measures the proportion of negative cases that are correctly classified as not AF versus all signals identified as not AF in the observed set.
- Positive Predictivity Value (PPV), also known as precision, is the proportion of signals correctly classified as AF versus the total number of AF in the set.

- Accuracy (Acc) measures the overall model's performance and is calculated as a ratio of correctly predicted observations (both AF and non-AF) to the total observations in the proportion.

However, the performance of algorithms submitted to the CinC Challenge 2017 was measured using the F1 score [5], a metric that provides a balanced view of the model's proficiency in accurately classifying both positive and negative instances. The F1 score is calculated as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

An empirical study on the performance evaluation of AF detection highlighted the F1 score's suitability as the premier metric for assessing AF detection algorithms [31]. This research confirms the effectiveness of the F1 score through a comprehensive examination of various data sets, promoting the value of the F1 score in giving a detailed understanding of how well an algorithm can accurately detect true positives while effectively minimizing false positives and negatives. This robust methodological approach makes a strong case for using the F1 score to evaluate how well algorithms detect atrial fibrillation.

3 Conclusions

This paper systematically reviewed the application of ML algorithms in the detection of AF using ECG data. Through the examination of databases, signal lengths, preprocessing techniques, and ML algorithms, it has been shown that ML can significantly enhance the precision and efficiency of AF detection. The analysis revealed a preference for certain public datasets, such as those provided by PhysioNet, and highlighted the absence of a universal standard for optimal signal length in AF detection, which can vary based on the objectives of the study and the capabilities of the ML algorithm used.

Preprocessing techniques for ECG signal analysis have evolved, with recent trends showing a shift towards models that require minimal or no preprocessing, like CNNs and RNNs. These advancements suggest a move towards more direct analysis of raw ECG signals, simplifying the detection process. Furthermore, the exploration of various ML algorithms, including deep learning models like CNNs and LSTMs, demonstrated their

advancement over traditional machine learning models due to their ability to process complex patterns within ECG signals more effectively. However, it should be also highlighted that no single best method for AF detection has emerged due to variations in ML methodologies and datasets.

The evaluation of the algorithms using metrics such as sensitivity, specificity, and the F1 score is important for determining their applicability in clinical settings. The findings support the use of the F1 score as a balanced measure of an algorithm's ability to accurately classify AF, which is vital for developing reliable diagnostic tools. However, even though the studies reviewed achieved high prediction scores, the results cannot be compared with each other because of the differences in the datasets and methodology used. Also, most of the studies evaluated the performance of ML algorithms using data from the same database used for training rather than unseen, real-life data. The research does not sufficiently clarify if the training and testing data consist of distinct patient groups. This distinction is critical because if a long ECG signal is segmented and portions of it are used for both training and testing, the algorithm's performance may appear artificially inflated, as it could learn and thus anticipate the heart's variability within an individual during training. It is imperative for future research to conduct cross-database testing, address data imbalance, and ensure that the training and testing datasets are truly independent to avoid inflated performance results.

The variance in signal lengths and preprocessing techniques, alongside the diverse array of machine learning algorithms, illustrates the complexity of achieving a standardized approach. Future studies should focus on refining ML models to improve their diagnostic accuracy on real-life data and establish benchmarks that enable the comparison of AF detection methods. Also, the next steps in ML for AF detection should strive for clinical validation. The ultimate goal is to integrate these ML algorithms seamlessly into clinical workflows, contributing to the early and precise detection of AF and improving patient outcomes. This endeavour requires a multi-faceted approach, combining the technical advancements in ML with rigorous clinical testing to establish the practical efficacy of these automated systems.

References

[1] "SinusRhythmLabels.png — wikipedia, the free encyclopedia," 2006. [Online; accessed 9-April-2024].

[2] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, June 2000.

[3] I. Olier, S. Ortega-Martorell, M. Pieroni, and G. Y. Lip, "How machine learning is impacting research in atrial fibrillation: implications for risk prediction and future management," *Cardiovascular Research*, vol. 117, no. 7, pp. 1700–1717, 2021.

[4] G. Moody, "A new method for detecting atrial fibrillation using rr intervals," *Proc. Comput. Cardiol.*, vol. 10, pp. 227–230, 1983.

[5] G. D. Clifford, C. Liu, B. Moody, H. L. Li-wei, I. Silva, Q. Li, A. Johnson, and R. G. Mark, "Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017," in *2017 Computing in Cardiology (CinC)*, pp. 1–4, IEEE, 2017.

[6] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE engineering in medicine and biology magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[7] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[8] C. Xie, Z. Wang, C. Yang, J. Liu, and H. Liang, "Machine learning for detecting atrial fibrillation from ecgs: Systematic review and meta-analysis," *Reviews in Cardiovascular Medicine*, vol. 25, no. 1, p. 8, 2024.

[9] N. A. Abdul-Kadir, N. M. Safri, and M. A. Othman, "Effect of ecg episodes on parameters extraction for paroxysmal atrial fibrillation classification," in *2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 874–877, IEEE, 2014.

[10] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam, "Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network," *Information sciences*, vol. 405, pp. 81–90, 2017.

[11] Z. Xiong, M. K. Stiles, and J. Zhao, "Robust ecg signal classification for detection of atrial fibrillation using a novel neural network. in 2017 computing in cardiology (cinc)," *IEEE. https://doi. org/10.22489/CinC*, vol. 138, 2017.

[12] R. S. Andersen, A. Peimankar, and S. Puthusserypady, "A deep learning approach for real-time detection of atrial fibrillation," *Expert Systems with Applications*, vol. 115, pp. 465–473, 2019.

[13] V. Gliner and Y. Yaniv, "An svm approach for identifying atrial fibrillation," *Physiological Measurement*, vol. 39, no. 9, p. 094007, 2018.

[14] N. Sadr, M. Jayawardhana, T. T. Pham, R. Tang, A. T. Balaei, and P. de Chazal, "A low-complexity algorithm for detection of atrial fibrillation using an ecg," *Physiological measurement*, vol. 39, no. 6, p. 064003, 2018.

[15] R. S. Andersen, E. S. Poulsen, and S. Puthusserypady, "A novel approach for automatic detection of atrial fibrillation based on inter beat intervals and support vector machine," in *2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 2039–2042, IEEE, 2017.

[16] R. Smisek, J. Hejc, M. Ronzhina, A. Nemcova, L. Marsanova, J. Kolarova, L. Smital, and M. Vitek, "Multi-stage svm approach for cardiac arrhythmias detection in short single-lead ecg recorded by a wearable device," *Physiological measurement*, vol. 39, no. 9, p. 094003, 2018.

[17] F. Plesinger, P. Nejedly, I. Viscor, J. Halamek, and P. Jurak, "Parallel use of a convolutional neural network and bagged tree ensemble for the classification of holter ecg," *Physiological measurement*, vol. 39, no. 9, p. 094002, 2018.
[18] K.-S. Lee, S. Jung, Y. Gil, and H. S. Son, "Atrial fibrillation classification based on convolutional neural networks," *BMC medical informatics and decision making*, vol. 19, pp. 1–6, 2019.
[19] S. Mousavi, F. Afghah, A. Razi, and U. R. Acharya, "Ecgnet: Learning where to attend for detection of atrial fibrillation with deep visual attention," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pp. 1–4, IEEE, 2019.
[20] S. Liaqat, K. Dashtipour, A. Zahid, K. Assaleh, K. Arshad, and N. Ramzan, "Detection of atrial fibrillation using a machine learning approach," *Information*, vol. 11, no. 12, p. 549, 2020.
[21] R. Kamaleswaran, R. Mahajan, and O. Akbilgic, "A robust deep convolutional neural network for the classification of abnormal cardiac rhythm using single lead electrocardiograms of variable length," *Physiological measurement*, vol. 39, no. 3, p. 035006, 2018.
[22] S. Hong, M. Wu, Y. Zhou, Q. Wang, J. Shang, H. Li, and J. Xie, "Encase: An ensemble classifier for ecg classification using expert features and deep neural networks," in *2017 Computing in cardiology (cinc)*, pp. 1–4, IEEE, 2017.
[23] M. Kropf, D. Hayn, D. Morris, A.-K. Radhakrishnan, E. Belyavskiy, A. Frydas, E. Pieske-Kraigher, B. Pieske, and G. Schreier, "Cardiac anomaly detection based on time and frequency domain features using tree-based classifiers," *Physiological measurement*, vol. 39, no. 11, p. 114001, 2018.
[24] M. Zabihi, A. B. Rad, A. K. Katsaggelos, S. Kiranyaz, S. Narkilahti, and M. Gabbouj, "Detection of atrial fibrillation in ecg hand-held devices using a random forest classifier," in *2017 Computing in Cardiology (CinC)*, pp. 1–4, IEEE, 2017.
[25] A. A. Laghari, Y. Sun, M. Alhussein, K. Aurangzeb, M. S. Anwar, and M. Rashid, "Deep residual-dense network based on bidirectional recurrent neural network for atrial fibrillation detection," *Scientific Reports*, vol. 13, no. 1, p. 15109, 2023.
[26] D. Kumar, S. Puthusserypady, H. Dominguez, K. Sharma, and J. E. Bardram, "An investigation of the contextual distribution of false positives in a deep learning-based atrial fibrillation detection algorithm," *Expert Systems with Applications*, vol. 211, p. 118540, 2023.
[27] F. Liu, C. Liu, L. Zhao, X. Zhang, X. Wu, X. Xu, Y. Liu, C. Ma, S. Wei, Z. He, *et al.*, "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection," *Journal of Medical Imaging and Health Informatics*, vol. 8, no. 7, pp. 1368–1373, 2018.
[28] D. Kumar, S. Puthusserypady, H. Dominguez, K. Sharma, and J. E. Bardram, "Cachet-cadb: A contextualized ambulatory electrocardiography arrhythmia dataset," *Frontiers in Cardiovascular Medicine*, vol. 9, p. 893090, 2022.
[29] H. B. Wong and G. H. Lim, "Measures of diagnostic accuracy: sensitivity, specificity, ppv and npv," *Proceedings of Singapore healthcare*, vol. 20, no. 4, pp. 316–318, 2011.
[30] W. Zhu, N. Zeng, N. Wang, *et al.*, "Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations," *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, vol. 19, p. 67, 2010.
[31] M. Gusev and M. Boshkovska, "Performance evaluation of atrial fibrillation detection," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 342–347, IEEE, 2019.

Formalizing IOTA Extended UTXO in Isabelle

Edvardas Dlugauskas, Karolis Petrauskas

Vilnius University, Faculty of Mathematics and Informatics
Didlaukio str. 47, Vilnius
edvardas.dlugauskas@mif.vu.lt, karolis.petrauskas@mif.vu.lt

Abstract. The IOTA Extended UTXO (IOTA EUTXO) model extends the UTXO blockchain to include features like smart contracts and non-fungible tokens. In this work, we show that the IOTA EUTXO model maintains the base correctness properties of the UTXO model while extending it with extra functionality. We achieve this by specifying and verifying the essential concepts of the base UTXO model and the extensions proposed by IOTA using the Isabelle proof assistant. The specification is designed to be modular and extensible, meaning it can be used as a foundation for further research of the UTXO and IOTA EUTXO models.

Keywords: IOTA, UTXO model, EUTXO model, formal verification, Isabelle, formal methods.

1 Introduction

A blockchain is a decentralized digital ledger that records transactions in a way that is transparent and immutable. It uses accounts and tokens to represent digital asset ownership or rights, allowing peer-to-peer transactions without a trusted third party. The ledger in a blockchain network is usually implemented in one of two ways: using the Unspent Transaction Output (UTXO) model or the Account model. In the UTXO model, used by blockchains like Bitcoin, the ledger is represented as a set of unspent transaction outputs, referred to as just outputs in short [1]. Transactions consume outputs from the ledger as inputs and generate new outputs. This makes it possible to verify and process transactions which use different outputs as inputs independently. Subsequently, the UTXO model is known for its ability to allow parallel transaction processing, which improves scalability [2]. In contrast, the Account model, adopted by blockchains such as Ethereum, simplifies the ledger to a set of account balances [3]. Transactions adjust these balances directly. While more straightforward, this model requires transactions for the same account to be processed sequentially, which limits scalability [4].

DOI: https://doi.org/10.15388/LMITT.2024.3

In the UTXO model, digital assets (tokens), are represented using immutable outputs. The outputs are owned by actors, who can perform transactions, such as sending some amount of tokens to another actor. Crucially, each transaction irreversibly consumes its input outputs, generating new outputs to represent the transferred tokens. Every actor's owned amount of tokens at a given time can be calculated by taking the sum of tokens in all of the outputs owned by the actor. In the UTXO model, the current state is a set of all of the unspent transaction outputs. In other words, the UTXOs form a directed acyclic graph, and the current state is the set of all of the leaves of this graph [2].

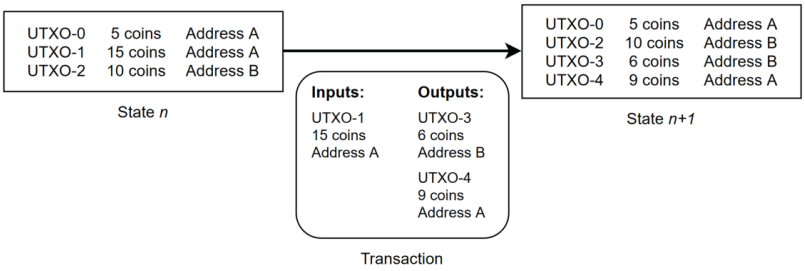


Figure 1. An illustration of a transaction in the UTXO model. In the UTXO model, the current state is a set of all of the unspent outputs.

Figure 1 illustrates a transaction in the UTXO model. In State *n*, we see three outputs: UTXO-0, UTXO-1, and UTXO-2, each associated with a certain number of tokens and an owning address (Address A or Address B). A transaction takes place where UTXO-1 is used as an input to create two new outputs: UTXO-3 and UTXO-4, which are then owned by Address B and Address A respectively. This transaction results in the subsequent State *n+1*, which now includes the two new outputs along with the unchanged UTXO-0 and UTXO-2 from the previous state.

The UTXO model supports some basic validation rules for output spending conditions. One of the most common conditions is the verification of ownership by the inspection of the digital signature on the output. However, there is a need for more programmable blockchain logic [5]. To address this, the concept of a smart contract can be used.

In general terms, a smart contract is a protocol for verifying and enforcing contracts on a blockchain [6]. A smart contract is stored on the distributed

ledger, it inspects the state of the ledger, maintains and modifies its internal state, and performs actions such as creating new transactions. Smart contracts rely on more complex validation rules that are not supported by the base UTXO model. Due to the complexities of implementing a smart contract, the more straightforward account model is usually used instead [7].

One way to introduce expressive smart contracts while maintaining the semantic simplicity of the UTXO model is to implement the validation logic on the outputs. Subsequently, as the UTXO model is stateless, a smart contract's transactions would be forced to include any state information in the outputs themselves, introducing complexity to the model [5].

IOTA is a blockchain based on the UTXO model that started out with the aim of powering high throughput applications with low-price transactions [8]. TIP-18[1] is a design document that describes extensions to the IOTA UTXO model to add support for features such as NFTs and smart contracts. We refer to this proposed model as IOTA Extended UTXO (IOTA EUTXO) model.

The IOTA EUTXO model extends the traditional UTXO model. The goal of the IOTA EUTXO is to add the functionality of smart contracts while maintaining the base UTXO model's advantages. This is achieved by appending additional data fields and extending the validation logic in the outputs. Thus, the EUTXO model allows for more complex transactions and behaviors without limiting the model's scalability [9].

Correctness is crucial to blockchain technologies as every processed transaction, whether correct or not, is permanent. This means that any oversight or vulnerability in the transaction processing logic can be impossible to revert [10]. Subsequently, the complexity of the changes proposed in the IOTA EUTXO design document raises the question of IOTA EUTXO model's correctness.

Formal methods are a set of techniques to accurately specify and verify software systems [11]. By applying formal methods to the verification of blockchain protocols and smart contracts, we can identify any incorrect behavior of the system at an early stage [12]. In the case of IOTA EUTXO model, formal methods can be used to prove the correctness of the proposed model [13].

---

1 https://github.com/lzpap/tips/blob/master/tips/TIP-0018/tip-0018.md

The formal verification of blockchain models or their smart contracts often involves using a proof assistant [14]. Isabelle is a collection of tools that allow formally verifying specifications using higher-order logic [15]. It is currently one of the more popular formal verification tools in academia due to its intuitive development environment and use of powerful provers. As such, Isabelle is a solid choice for specifying and verifying both the base UTXO model and IOTA's EUTXO model.

While there have already been attempts to formalize the UTXO model, some of them using Isabelle, the results of these attempts are difficult to reuse for formalizing the IOTA EUTXO. Many of the formalizations, such as the Cardano UTXO specification[2], are missing an accompanying paper, which complicates further analysis of the specification, and do not explicitly consider the possibilities of extending the model. Other specifications, such as the mathematical specification of the UTXO model by Gabbay et al., use manual proofs [16]. Furthermore, the abstract nature of the models does not address the concern of their real-life applicability. Thus, a new formalization of the UTXO and IOTA EUTXO is required.

In this paper, we formalize the IOTA EUTXO model using Isabelle. We demonstrate a way to represent the essential entities and properties of the UTXO model in a modular way using Isabelle's syntax, including the locale construct. We then build on the UTXO model by formalizing a subset of the IOTA EUTXO's functionality. We show that the IOTA EUTXO model maintains the base UTXO model's properties while allowing for more complex workflows. By splitting the specification into implementation and abstract parts, we ensure that it is both theoretically sound and practically feasible. Our work provides a solid foundation for future UTXO and IOTA EUTXO model research by creating reusable components in Isabelle.

2 Formalizing the UTXO Model in Isabelle

Our formalization of the UTXO model in Isabelle uses a two-layered approach, differentiating an abstract and a implementation specification. The abstract layer represents the core properties and operations of outputs, transactions, and the ledger without tying them to specific data types – relationships and properties are represented using generic predicates

[2] https://github.com/input-output-hk/cardano-ledger-high-assurance/blob/master/Isabelle/UTxO/UTxO.thy

instead. This allows for the verification of the UTXO model's properties in a generic manner, ensuring any valid concrete implementation will inherit these properties.

When reasoning about the abstract model, we found the Isabelle locale construct to be very useful in ensuring a modular and extendable specification. A locale in Isabelle is a collection of parameters and assumptions that provide a context for proving theorems. To be more precise, locales are a way to define abstract contexts and structures, which can be instantiated later with specific types, functions, or relations. They provide a mechanism to reason about abstract properties and assumptions, prove theorems in a generic context and reuse the results in specific instances.

$$\forall x_1, \ldots, x_n.[(A_1; \ldots; A_m) \Rightarrow C] \tag{1}$$

In Eq. 1 parameters x_1 to x_n are fixed, assumptions A_1 to A_m are made, and the conclusions C are implied. When writing Isabelle code, C would correspond to the proofs for lemmas and theorems that can be proven inside the context of the locale. A locale can then be instantiated by satisfying its parameters and assumptions using a specific type by using the interpretation mechanism, which allows us to reuse the proven properties of the locale.

The abstract model focuses on the UTXO model's essential entities and properties. Using Isabelle locales, we model basic entities like outputs, transactions, and the ledger, each with its own essential properties and operations. For instance, the *basic_output* locale guarantees that each output possesses a non-zero amount, while the basic transaction locale ensures the conservation of total token amount in a transaction.

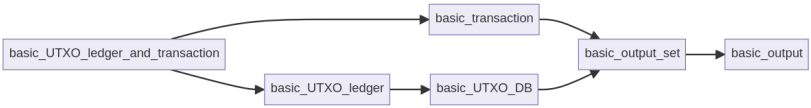


Figure 2. The locale structure in the abstract UTXO model. Six distinct locales are represented as nodes, the relationships between them are depicted with directed edges.

Figure 2 depicts the hierarchical structure of locales in the abstract UTXO model as used in formal verification within the Isabelle proof assistant framework. It illustrates how six distinct locales interconnect to model the UTXO model, with each node representing a locale and the edges indicating the dependency and extension relationships between them. The fun-

damental locale is *basic_output*, which defines individual UTXOs. It is used by *basic_output_set*, representing a collection of UTXOs. This, in turn, is referenced by the *basic_transaction_locale*, which models the transaction mechanism. The *basic_UTXO_ledger* locale relies on *basic_UTXO_DB*, which itself references *basic_output_set*. Both *basic_transaction* and *basic_UTXO_ledger* are used by the *basic_UTXO_ledger_and_transaction* locale, which encapsulates the ledger's state and a valid transaction, allowing us to reason in terms of the current and the subsequent ledger's state.

In contrast, the implementation layer offers a concrete example implementation of the UTXO ledger, defining specific data types for outputs, transactions, and other essential entities, as well as functions and predicates to model ledger updates and transaction validity. By mapping these concrete types to their abstract counterparts and verifying that the abstract model's assumptions still hold, we demonstrate that the implementation adheres to the desired properties of the UTXO model.

By using Isabelle's interpretation mechanism, we link the implementation model's concrete entities to the abstract model's locales and assumptions. This validates the implementation against the abstract specification and ensures the inheritance of all proven properties from the abstract model. Thus, we establish the correctness of our UTXO model implementation, by ensuring it is both theoretically sound and practically viable.

3 Formalizing the IOTA Extended UTXO Model in Isabelle

The IOTA EUTXO design document describes several new output types such as alias output and foundry output. An alias output is an output representing smart contract invocation chain accounts that can process requests and transfer funds. A foundry output is an output that contains the state of and manages user-defined native tokens [17]. To support these new output types, the ledger has to have some characteristics of a state machine.

For an output to function as a state machine, the state of the output must be moved forward when it is consumed as an input. In the UTXO model, the input outputs are essentially burned and only the value amount is distributed among the outgoing outputs. Subsequently, IOTA proposes an extension to the validator called a chain constraint. The chain constraint allows the transfer of the output state machine state encoded in the new additional fields on the output across transactions. The alias output and foundry output utilize this chain constraint to transfer the states of the alias

state machine and foundry state machine respectively. The chain constraint describes validation rules that ensure that the state is created, modified, and deleted in a correct manner. For example, the chain constraint ensures that once an alias is created, it has a continuous existence across the ledger until explicitly deleted and removed from the ledger.

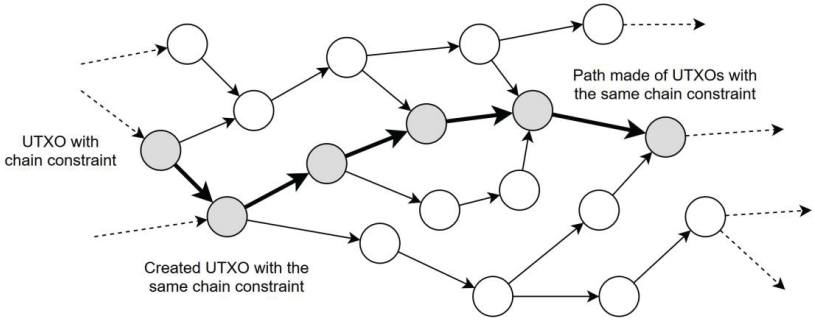


Figure 3. A path of consumed outputs with the same chain constraint forms a chain. Arrows indicate the creation of an output and the circles are the transactions; gray circles are transactions that contain a chain constraint.

Figure 3 visualizes the concept of a transaction chain that uses the chain constraint. In this representation, each circle symbolizes a transaction, with the gray circles representing transactions that include an output with a chain constraint. The arrows between the circles represent the use of an output from one transaction in the next. A path made up of outputs linked by the same chain constraint is highlighted. This chain begins with the creation of an output with a chain constraint and terminates when the output containing this constraint is spent without creating a new one, which ends the sequence.

Our formalization of the UTXO model in Isabelle uses the same two-layered approach used for the base UTXO model, with an abstract and a implementation specification. The IOTA EUTXO formalization builds upon the locales of the basic UTXO model.

In the abstract model, the details of the alias and foundry outputs are added. Notably, the base output and the additional state fields in an alias or foundry output are independent in terms of the operations and the properties of the UTXO model. Properties and operations from the base model only reference fields of the base model. Analogously for alias and

foundry outputs. Thus, we have modeled them as separate ledgers. For example, an alias ledger is an image of the blockchain ledger containing only the alias parts of the outputs; an alias ledger contains details of only the alias outputs and validates logic related only to alias output functionality. This approach ensures that our specification remains modular and reusable.

In the implementation model, we have opted to use a single ledger to represent all of the output types, which better mimics a possible real-world implementation. Instead, we define the ledger as a set of a sum type with three possible constructors: basic output, alias output and foundry output. The basic output contains only the base output fields, while alias and foundry contain both the basic output fields and the alias and foundry fields respectively. This allows us to map the implementation to the abstract specification – an alias ledger is just the image of the implementation ledger which takes all of the alias field parts of the outputs in the ledger.

Using the Isabelle locale interpretation feature, we link the IOTA EUTXO implementation model's concrete entities to both the base output and the IOTA EUTXO abstract model's locales and assumptions. Subsequently, we establish the correctness of the implementation model in the context of both the base UTXO and the IOTA EUTXO models' invariants and properties.

4 Formalization Results

In our work, we not only formalized the models, but also verified some of their properties using Isabelle automated provers.

The UTXO model has several essential properties that we have verified:

- Constant Supply: the sum of unspent outputs in the ledger must be constant.
- Unspent Output Consumption: an output can be consumed only if it is a part of the current ledger state and this output will not be present in the subsequent ledger state.
- No Double Spending: an output can only be consumed by a single transaction.

In the IOTA EUTXO model, we have verified all of the base UTXO model's properties in addition to the chain constraint for the alias output:

- Continuity of Alias (Chain Constraint): once an alias is created, it has a continuous existence across the ledger until explicitly deleted and removed from the ledger.

The proof process for verifying these properties in Isabelle involved several steps. For each property we aimed to verify, we started by formulating a theorem definition for it using Isabelle. This required translating informal descriptions of UTXO model's behavior into precise, logical statements using Isabelle's syntax. We then used Isabelle's automated proof search tools and manual proof strategies to construct a proof for each theorem. Finally, we used Isabelle's automated provers to ensure that the proof was sound.

To demonstrate the verification of one specific property in more detail, let's consider the constant supply property of the UTXO model. This property ensures that the total sum of tokens across all unspent outputs remains unchanged by the application of transactions, assuming no new tokens are minted or existing tokens are destroyed outside of transactions.

We first formally defined the *sum_amount* function in Isabelle, which calculates the total sum of tokens in a given set of outputs. The *constant_supply* theorem was then stated as:

$$sum\_amount\ DB = sum\_amount\ (apply\_transaction\ DB\ tx) \tag{2}$$

In Eq. 2 we assume that *tx* is a valid transaction in the ledger *DB*.

We then constructed a proof by interacting with Isabelle's automatic proof search functionality. The proof uses the subproofs for the facts that the amount of tokens in the inputs and outputs of a valid transaction is the same, and that, in terms of tokens, applying a transaction is equivalent to subtracting the amount of tokens in the inputs and adding the amount of tokens in the outputs. Finally, the Isabelle automated provers verified our proof to demonstrate its soundness.

5 Conclusions

We aimed to formalize the IOTA EUTXO model by utilizing the Isabelle formal verification tool and verify it using Isabelle's automated prover. Our formalization showed that the IOTA EUTXO model not only retains the base correctness properties of the base UTXO model including no double spending, constant supply, and unspent output consumption, but also supports additional properties which are specified in the IOTA EUTXO design document, such as the chain constraint.

We have used a two-layered approach in our formalization to ensure that our model is both theoretically robust and practically applicable. The

abstract layer allowed us to verify the UTXO model's core properties in a generic way, while the implementation layer provided a concrete example that adheres to the verified properties, demonstrating the practical viability of our formalization.

We have presented a model that is not only modular but also extensible. Our approach to formalization, which uses Isabelle locales, provides flexibility in specifying the model by allowing easier future extensions or modifications. This is crucial for keeping the model relevant as distributed ledger technologies continue to evolve. We believe that this approach offers a good foundation for further research and developments in the field of UTXO blockchain technologies.

References

[1] Almeida, J. B., Frade, M. J., Pinto, J. S., and De Sousa, S. M. (2011). Rigorous software development: an introduction to program verification, volume 1. Springer.

[2] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., and Zanella-Béguelin, S. (2016). Short paper: Formal verification of smart contracts. In Proceedings of the 11th ACM Workshop on Programming Languages and Analysis for Security (PLAS), in conjunction with ACM CCS, pages 91–96.

[3] Brünjes, L. and Gabbay, M. J. (2020). UTxO- vs account-based smart contract blockchain programming paradigms. In International Symposium on Leveraging Applications of Formal Methods, pages 73–88. Springer.

[4] Chakravarty, M. M., Chapman, J., MacKenzie, K., Melkonian, O., Jones, M. P., and Wadler, P. (2020). The extended UTXO model. In International Conference on Financial Cryptography and Data Security, pages 525–539. Springer.

[5] Chakravarty, M. M. T., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Peyton Jones, M., Vinogradova, P., and Wadler, P. Native custom tokens in the extended UTXO model. In Margaria, T. and Steffen, B., editors, Leveraging Applications of Formal Methods, Verification and Validation: Applications, Lecture Notes in Computer Science, pages 89–111. Springer International Publishing.

[6] Delgado-Segura, S., Pérez-Sola, C., Navarro-Arribas, G., and Herrera-Joancomartí, J. (2018). Analysis of the bitcoin UTXO set. In International Conference on Financial Cryptography and Data Security, pages 78–91. Springer.

[7] Fatkina, A., Iakushkin, O., Selivanov, D., and Korkhov, V. (2019). Methods of formal software verification in the context of distributed systems. In International Conference on Computational Science and Its Applications, pages 546–555. Springer.

[8] Gabbay, M. J. (2022). Algebras of UTxO blockchains. Mathematical Structures in Computer Science, pages 1–56.

[9] Liu, Y.-C., Fang, J., and Liang, J.-W. (2019). Account-wise ledger: A new design of decentralized system. Github. https://github.com/ECS-251-W2020/final-project-triple-l-group/blob/master/Thesis/Account-Wise%20Ledger.pdf

[10] Melkonian, O. (2019). Formalizing Extended UTxO and BitML Calculus in Agda. Master's thesis. Utrecht University Student Theses Repository Home. https://studenttheses.uu.nl/bitstream/handle/20.500.12932/32981/thesis.pdf
[11] Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). Isabelle/HOL: a proof assistant for higher-order logic, volume 2283. Springer Science & Business Media.
[12] Popov, S. and Lu, Q. (2019). Iota: feeless and free. IEEE Blockchain Technical Briefs.
[13] Ribeiro, M., Adão, P., and Mateus, P. (2020). Formal verification of ethereum smart contracts using Isabelle/HOL. In Logic, Language, and Security, pages 71–97. Springer.
[14] Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., and Wang, F.-Y. (2019). Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(11):2266–2277.
[15] Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014):1–32.
[16] Zhang, J., Tian, R., Cao, Y., Yuan, X., Yu, Z., Yan, X., and Zhang, X. (2021). A hybrid model for central bank digital currency based on blockchain. IEEE Access, 9:53589–53601.
[17] Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress), pages 557–564. IEEE.

Kokybės funkcijų sklaidos metodas sistemos dekomponavimui įgyvendinti

Gytis Grigonis

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Didlaukio g. 47, LT-08303 Vilnius
gytis.grigonis@mif.stud.vu.lt

Santrauka. Sudėtingos sistemos dekomponavimas, suskaidantis ją į silpnai sukibusias dalis, kad būtų lengviau tą sistemą sukurti, yra esminis sėkmę nulemiantis dalykas. Šiai problemai spręsti siūlome taikyti kokybės sklaidos metodą, leidžiantį išlaikyti ryšį tarp aukštesnio lygmens dalių ir jas sudarančių komponentų, o taip pat nustatyti, kuriam komponentui priskirti nagrinėjamą elementą dekomponavimo nevienareikšmiškumo atveju. Tai argumentais autoriaus poziciją grindžiantis straipsnis (angl. *position paper*), kuriame pateikiami sudėtingų sistemų dekomponavimo tyrimo rezultatai.

Raktiniai žodžiai: Sistemų dekomponavimas, sudėtingos informacinės sistemos, kokybės funkcijų sklaidos metodas, komponentas, klasterizavimas.

1 Įvadas

Šiuolaikinės informacinės sistemos (IS) plečiasi ir tampa vis sudėtingesnės. Jos apibrėžiamos kaip integruotas komponentų rinkinys, skirtas duomenims rinkti, saugoti ir apdoroti bei teikti ne tik informaciją ir žinias, bet ir skaitmeninius produktus. Sistemos dekomponavimas, suskaidantis ją į silpnai sukibusias dalis, yra esminis kuriamos sistemos sėkmę nulemiantis sprendinys.

Sistemos dekomponavimas dažniausiai nagrinėjamas sistemos projektavimo kontekste. Didelių sistemų sudėtingumui valdyti, dekomponuojant jas į dalis, mokslinėje literatūroje skiriamas kiek mažesnis dėmesys, o struktūros projektavimo matrica [1] yra viena populiariausių tam taikomų technikų. Skiriamos įvairios jos pagrindu sukurtų metodų variacijos; o sistemos dalys nustatomos klasterizavimo būdu [2]. Tai reiškia, kad matricos eilutėse ir stulpeliuose nurodomi tie patys elementai, kas neleidžia apimti ryšio tarp aukštesnio lygmens dalių ir jas sudarančių komponentų. Tam išspręsti šiame darbe siūloma taikyti kokybės funkcijų sklaidos (KFS) metodą. Be to, KFS matricos pildymo taisyklės įgalina spręsti klasterių formavimo proble-

DOI: https://doi.org/10.15388/LMITT.2024.4

mą tais atvejais, kai reikia nustatyti kuriam iš kelių galimų klasterių priskirti nagrinėjamą elementą.

Kituose straipsnio skyriuose pateikiama argumentuota autoriaus pozicija.

2 Susiję darbai

Struktūros projektavimo matricos (SPM) pagrindu sukurtas metodas, yra naudojamas sistemų dekomponavimo procese, siekiant modeliuoti, analizuoti ir valdyti sistemų sudėtingumą [1, 2]. Naudodami SPM, inžinieriai gali užtikrinti informacinių ir programų įrangos sistemų moduliškumą, t. y., identifikuoti tarpusavyje susijusius elementus, kuriuos galima sugrupuoti į komponentus. Grupavimas vykdomas identifikuojant galimus klasterius. Informacinių sistemų kūrimo procese tai padeda veiksmingai vizualizuoti programinės įrangos komponentų santykius ir valdyti programinės įrangos architektūros sudėtingumą.

Kiekviena matricos dalis nurodo ryšio tarp komponento eilutėje ir komponento stulpelyje buvimą arba tipą. Ryšiai gali reikšti duomenų srautus, valdymo srautus, funkcines priklausomybes ar kitokias priklausomybes. Matrica dažniausiai būna dviejų tipų: dvejetainė (nurodanti priklausomybės buvimą ar nebuvimą) arba svertinė (pateikianti daugiau informacijos apie priklausomybės stiprumą ar svarbą). SPM plačiai taikomas, ypač inžinerinio projektavimo srityse. Tačiau klasikinis jo variantas netinkamas sistemos savybėms nuleisti žemyn, kitaip tariant, nepalaiko ryšių tarp skirtingų lygmenų komponentų, reikiamų dekomponavimo procese.

Kokybės funkcijų sklaidos (KFS) metodas, naudojamas produktų ir paslaugų kūrimo procesuose, siekiant užtikrinti, kad sukurtos sistemos tiksliai atitiktų klientų poreikius ir kokybės reikalavimus [3]. Kalbant metodo terminais – tai sistemingas „kliento balso“ vertimas į žemesnio lygmens veiksmus, reikalingus patenkinti klientų poreikius. Šis metodas buvo taikytas ir komponentams identifikuoti [4], tačiau, panašiai kaip SPM, tam buvo nustatomi eilutėse esančių savybių ir stulpelyje išvardintų operacijų klasteriai (objektai objektinės paradigmos prasme).

3 Rezultatai

Kokybės funkcijų sklaidos (KFS) metodas [5] nuo jo sukūrimo 1966 m. Japonijoje, plačiai naudojamas įvairių sričių inžinerijoje, nes leidžia sumažinti

kuriamos sistemos perdarymų skaičių. Jo įvairios modifikacijos taikytos ir programų sistemų kurti.

Informacinės sistemos dekomponavimas gali būti įgyvendintas KFS pagrindinės matricos pagrindu. Kairiajame stulpelyje nurodomos IS, kaip monolitą aprašančios savybės, o stulpeliuose tos savybės išreikštos žemesnio lygmens terminais. Tada matricoje sužymimos priklausomybės, kur „+“ žymi silpną sąryšį, o „++“ – stiprų. Kitas žingsnis – matricos pertvarkymas, siekiant suformuoti galimus savybių klasterius. Jei visos arba beveik visos aukštesnio lygmens savybės turi ryšį su viena žemesnio lygmens savybe, ši žemesnio lygmens savybė sudaro atskirą klasterį. Kitaip sakant, toks atvejis reiškia, kad aukštesnio lygmens savybė negali būti „išbarstyta“ po kelis komponentus (pvz., sistemos apsauga, kuri kaip atskiras aspektas lokalizuojamas viename komponente). 1 pav. pateiktame pavyzdyje matoma, kad pertvarkius matricą, gauname tris komponentus, kurie yra sužymėti atitinkamai mėlyna, oranžine ir violetine spalvomis. S3 ir Ž7 susietos stipriu ryšiu, todėl savybę Ž7 priskiriame mėlynajam komponentui.

IS savybė	IS žemesnio lygmens savybės						
	Ž1	Ž2	Ž3	Ž4	Ž5	Ž6	Ž7
S1	++		+	+	+		
S2		+		+		+	+
S3	+			+	+		++
S4			+	+	+		
S5		+		+			+

pertvarkymas

IS savybė	IS žemesnio lygmens savybės						
	Ž1	Ž3	Ž5	Ž7	Ž2	Ž6	Ž4
S1	++	+	+				+
S4		+	+				+
S3	+		+	++			+
S2					+	+	+
S5				+	+		+

1 pav. KFS panaudojimas sistemos dekomponavimui

4 Išvados

Kuriant dideles sudėtingas informacines sistemas reikia ne tik jas dekomponuoti į dalis, bet ir kartu spręsti trasavimo problemą – sekti reikalavimo (sistemos savybės) gyvavimą nuo reikalavimo suformavimo iki įgyvendinimo sistemoje. Tam rekomenduojama taikyti adaptuotą kokybės funkcijų sklaidos metodą – savybių klasterizavimo pagrindu suformuojami komponentai, o pats matricos pavidalas užtikrina galimybę „nepamesti“ ryšių tarp skirtingų lygmenų artefaktų.

Tolesnio tyrimo turinys gali apimti dekomponavimą, kuriame atsižvelgiama ne tik į ryšius tarp skirtingų lygmenų, bet ir tų ryšių įvairovę bei stiprumo laipsnį; taip pat siekiama mažinti komponentų sukibimą.

Literatūra

[1] Steward, D. V. (1981) The Design Structure System: a method for managing the design of complex systems. IEEE Transactions on Engineering Management, 28(3), 1981, S. 71-74.

[2] Browning, T. R. (2016) Design Structure Matrix extensions and innovations: a survey and new opportunities. IEEE Transactions on Engineering Management, 63(1), 27–52.

[3] Lai-Kow Chan, M.-L. W. (2002) Quality function deployment: a literature review. European Journal of Operational Research. 2002, tomas CXLIII, 463–497. Prieiga per internetą: https:// www . sciencedirect . com / science / article / abs / pii /S0377221702001789.

[4] Lamia, W.M. (1995). Integrating QFD with object oriented software design methodologies. The 7th Symposium on Quality Function Deployment, Novi, Michigan, 18 p.

[5] Maritan, D. (2015). Quality Function Deployment (QFD): definitions, history and models. In: Practical Manual of Quality Function Deployment. Springer, Cham, 1-34.

Artimo gamtai miškininkavimo modelių vystymas naudojant miškininkavimo sprendimų paramos sistemą Heureka

Dalius Gudeika[1], Gintautas Mozgeris[2]

[1] Vytauto Didžiojo universitetas, Informatikos fakultetas, Universiteto g. 10-202, 53361 Akademija, Kauno rajonas
[2] Vytauto Didžiojo universitetas, Žemės ūkio akademija, Miškų ir ekologijos fakultetas, Studentų g. 11-443, LT-53361, Akademija, Kauno rajonas
dalius.gudeika@vdu.lt

Santrauka. Siekiant geriau suvokti sistemos Heureka funkcionalumą, ypač įvairių miškininkavimo scenarijų nustatymo ypatumus, buvo sumodeliuota Lietuvos miškų raida. Visais atvejais naudoti visos Lietuvos miško išteklių duomenys bei Lietuvos miškų augimas buvo modeliuojamas pagal modelius, taikomus geografiniu požiūriu artimiausiame Švedijos regione. Nepaisant tam tikrų metodinių problemų, eksperimento metu diskutuota prielaida, kad Lietuvos miško ūkis yra labiau draugiškas aplinkai, tačiau tuo pačiu mažiau efektyvus ekonominiais aspektais nei Švedijos.

Raktiniai žodžiai: Miškininkavimas, funkcionalumas, miškų raida, Heureka.

1 Įvadas

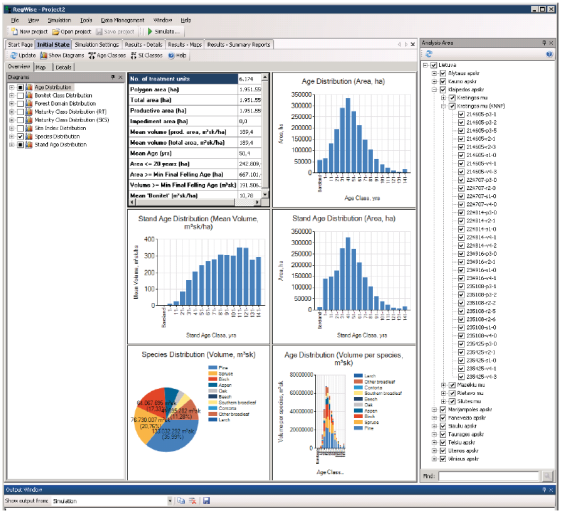
Miškininkavimo sprendimų priėmimo paramos sistema Heureka yra universalus įrankis [1-3], ilgamečių Švedijos miškininkų pastangų rezultatas, kurio naudojimas plečiamas ir kitose šalyse. Šią sistemą testavimui pasirinkome turėdami omenyje jos atvirumą, plėtojimo galimybes bei Lietuvos ir Švedijos Nacionalinių miškų inventorizacijų tam tikrą panašumą. Sistemos Heureka eksperimentas vykdytas glaudžiai bendradarbiaujant su jos autoriais iš Švedijos žemės ūkio mokslų universiteto. Šio tyrimo tikslas yra įvertinti miškininkavimo sprendimų paramos sistemos Heureka galimybes naudoti netipiniams Lietuvoje miškininkavimo modeliams vystyti.

2 Rezultatai

Eksperimento metu į sistemos Heureka duomenų bazes buvo importuoti 1998–2002 bei 2011–2015 metų Lietuvos Nacionalinės miškų inventorizaci-

DOI: https://doi.org/10.15388/LMITT.2024.5

jos (NMI) ciklų duomenys. Sistemoje Heureka yra pateikiamos gausios pagalbinės priemonės objektui, kurį numatoma nagrinėti, apibūdinti (1 pav.).

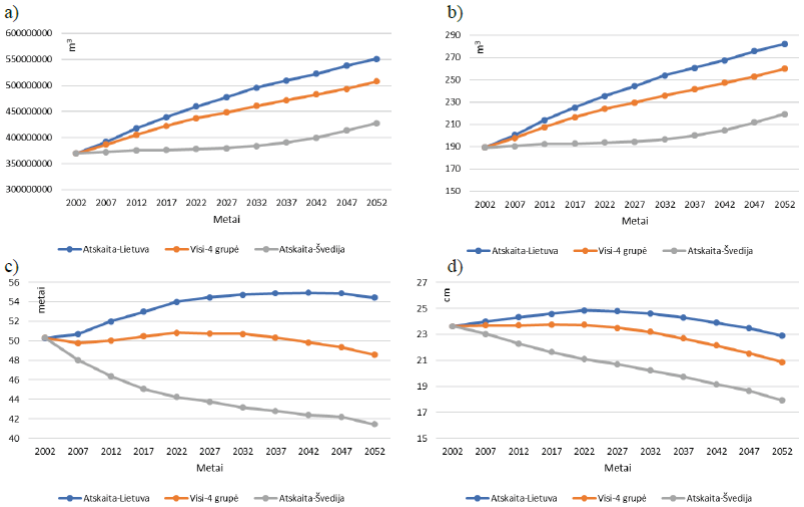


1 pav. Sistemos Heureka modulio *RegWise* vartotojo sąsaja, įkėlus Lietuvos NMI duomenis.

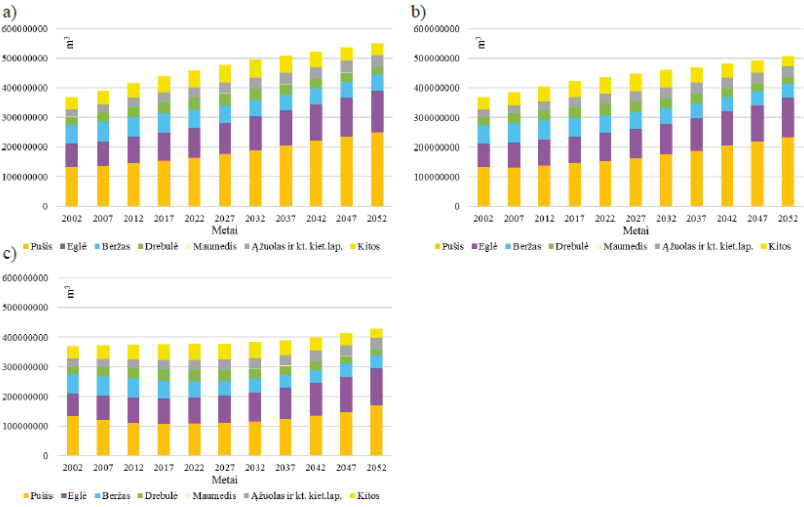
Modeliavimo rezultatai, gauti naudojant sistemą Heureka bei Lietuvos NMI duomenis yra pateikiami pirmiausia siekiant iliustruoti galimos gauti informacijos turinį. Konkretūs rodiklių dydžiai ar jų kaita per modeliuojamą laikotarpį yra nevertintini dėl Švedijoje ir Lietuvoje naudojamų modelių bazės skirtumų. Eksperimente pirmiausia siekta susipažinti su sistemos Heureka funkcionalumu bei jos potencialu, atitinkamai išvysčius, naudoti kaip Lietuvos Nacionalinės miškų inventorizacijos informacinės sistemos scenarijų modeliavimo posistemės (NMIIIS SMP) pagrindą, tačiau nekelta uždavinio sumodeliuoti vieną ar kitą šalies miškų raidos variantą. 2 pav. yra iliustruojama sumodeliuota kai kurių taksacinių rodiklių raida per 50 metų.

Pastebėtina, kad modeliuojant yra gaunami mažesni medynų tūriai, nei deklaruojama oficialiose miško išteklių statistikose (per pirmus modeliavimo žingsnius) bei naudojant kitas miškininkavimo scenarijų modeliavimo sistemas. Beje, skirtumai yra didesni, kuo daugiau miškininkavimo scenarijų apibūdinime yra su Švedijos miškininkyste susijusių nustatymų. Tačiau pažymėtina, kad sistema Heureka numatytuoju atveju pateikia gausią miško

išteklius apibūdinančią informaciją. Taksaciniai rodikliai apibendrinami pagal vyraujančias medžio rūšis (3 pav.), sortimentus.



2 pav. Kai kurių Lietuvos miškų taksacinių rodiklių raida, sumodeliuota naudojant sistemą Heureka; a) bendras medynų tūris, b) vidutinis medynų tūris, c) vidutinis medynų amžius ir d) vidutinis medynų skersmuo.



3 pav. Bendras medyno tūris, sumodeliuotas naudojant sistemą Heureka, pagal vyraujančią medžio rūšį; a) „Atskaita-Lietuva“, b) „Visi-4 grupė“ ir c) „Atskaita-Švedija“.

3 Išvados

Pasitelkus standartines sistemos Heureka bei darbo metu sukurtas priemones, pademonstruotas Lietuvos Nacionalinės miškų inventorizacijos duomenų importas į sistemą Heureka. Lietuvos Nacionalinės miškų inventorizacijos duomenims paruošti importui į sistemą Heureka yra sukurta MS Access grindžiama pagalbinė priemonė.

4 Padėka

Tyrimą finansuoja Europos Sąjunga (projekto Nr. [S-ST-23-224]) pagal sutartį su Lietuvos mokslo taryba (LMTLT).

Literatūra

[1] Mowrer, H. T., *Uncertainty in natural resource decision support systems: Sources, interpretation, and importance*, Comput. Electr. Agric., 2000, 27, 139–154. doi: 10.1016/S0168-1699(00)00113-7.

[2] Swedish University of Agricultural Sciences [SLU] (2022a). Heureka Wiki. Available online at: https://www.heurekaslu.se/wiki/Heureka_Wiki (accessed April 17, 2024).

[3] Borges, J. G., Nordström, E. M., Garcia-Gonzalo, J., Hujala, T., Trasobares, A. (eds), *Computer-based tools for supporting forest management. The experience and the expertise worldwide*, Umeå: Swedish University of Agricultural Sciences, 2014, 503.

Išgyvenamumo modelių taikymas personalo kaitai prognozuoti

Vilius Kavaliauskas

Vilniaus universitetas, Taikomosios matematikos institutas
Naugarduko g. 24, Vilnius
vilius.kavaliauskas@mif.stud.vu.lt

Santrauka. Personalo stabilumas yra itin svarbus įmonės sėkmės komponentas. Suprasti, kas labiausiai įtakoja darbuotojų kaitą, yra dažnai (ir prasmingai) darbdavio keliamas tikslas. Nors įprastai tam pasitelkiami klasikinės statistikos sprendimai, jie nebūtinai yra geriausias pasirinkimas. Šiame darbe standartiniam duomenų rinkiniui pritaikyti ir palyginti trys išgyvenamumo analizės metodai. Nustatyta, jog atsitiktiniai išgyvenimo miškai šiuo atveju veikia geriausiai.

Raktiniai žodžiai: personalo kaita, išgyvenamumo analizė, mašininis mokymasis.

1 Įvadas

Darbovietės personalo stabilumas yra svarbus kompanijos sėkmės komponentas. Darbdavio užduotis yra ne tik rasti aukštą potencialą turinčius individus, tačiau ir žinoti esminius faktorius, didinančius nepasitenkinimo ir, galiausiai, išėjimo riziką.

Šio uždavinio sprendimo metodų yra pakankamai. Logistinė regresija, Naivusis Bajeso, atsitiktinių miškų ar atraminių vektorių klasifikatoriai [1] dažnai minimi kaip galimi klasifikavimo metodai. Tačiau naudojant šiuos metodus neatsižvelgiama į laiko įtaką duomenims, kadangi kiekviena duomenų eilutė laikoma atskiru nepriklausomu stebėjimu [3].

Tokio kompromiso nereikia pritaikius išgyvenamumo analizės metodus, kai modeliuojamas laikas ir cenzūruoti stebėjimai naudojami kaip papildoma informacija. Tai nėra naujovė nei statistikos, nei darbuotojų kaitos uždavinio kontekste [3, 7]. Visgi, išgyvenamumo analizė nėra dažnai minima kaip galima personalo kaitos analizės alternatyva.

Šio darbo tikslas yra palyginti klasikinių ir inovatyvių išgyvenamumo analizės metodų taikymą darbuotojo išėjimo iš darbo prognozavimui.

DOI: https://doi.org/10.15388/LMITT.2024.6

2 Duomenys

Naudojamas laisvai prieinamas *Edward Babushkin* pateiktas realus duomenų rinkinys[1] apie 1129 darbuotojus iš įvairių pramonės sričių. Jame pateikti lytis, amžius, užmokesčio tipas, keliavimo į darbą būdas, įsidarbinimo šaltinis ir Didžiojo Penketo (angl. *Big Five)* asmenybės bruožų[2] įvertinimai. Duomenys išskaidyti į apmokymo ir testavimo aibes santykiu 7:3.

3 Metodai

Pirmasis metodas – parametrinė AFT (angl. *Accelerated Failure Time*) regresija [6]. Grafiniam tinkamumui nustatyti naudojamas sukurtas programinis įrankis, leidžiantis nustatyti tinkamus parametrinius skirstinius. Patikrinus eksponentinį, Veibulo, loglogistinį bei lognormalųjį skirstinius, akivaizdžiai netinka tik pastarasis. Tada skirstinių tinkamumas tikrinamas tikėtinumų santykio kriterijumi. Nustatyta, jog tinka Veibulo skirstinys.

Kitas naudojamas metodas – Kokso semiparametrinė proporcingųjų rizikų (angl. *Proportional Hazards*, PH) regresija [5]. Tai dažniausiai naudojamas ir geriausiai žinomas išgyvenamumo analizės modelis. Po stratifikavimo, proporcingųjų rizikų prielaidą modelis tenkina.

Trečias taikytas metodas – atsitiktiniai išgyvenimo miškai (angl. *Random Survival Forests*, RSF) [3]. Tai atsitiktinių miškų modifikacija, kur kiekviename medyje siekiama atskirti kuo labiau išgyvenimo charakteristika besiskiriančius individus. Esminiai hiperparametrai – medžių skaičius bei skaidymo taisyklė. Nustatyta, jog optimalią paklaidą fiksavo 500 medžių turintis miškas su lograngine skaidymo taisykle.

4 Rezultatai

Modelių palyginimui skaičiuojamas konkordancijos koeficientas [2]. Kuo jis arčiau 1, tuo modelis veikia tiksliau. Tiek apmokymo, tiek testavimo aibėje geriausiai pasirodo RSF modelis (1 lentelė). Tiesa, konkordancija testavimo aibėje pastebimai sumažėja.

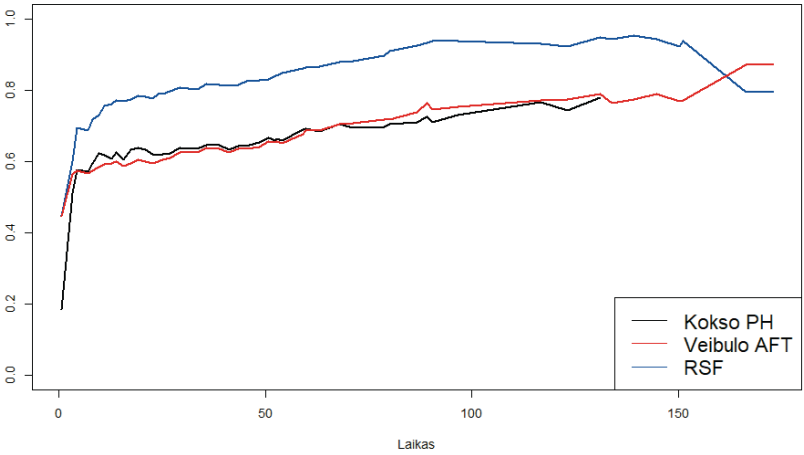
---

1 Viešai prieinamas Kaggle bei autoriaus tinklalapyje.
2 Emocinio stabilumo, ekstraversijos, atvirumo patirtims, sukalbamumo, sąmoningumo

1 lentelė. Konkordancijos indeksas apmokymo ir testavimo aibėse.

	Apmokymo aibė	Testavimo aibė
Veibulo AFT	0,659	0,565
Kokso PH	0,636	0,548
RSF	0,829	0,622

Apmokymo aibei nubraižomas dinaminis AUC (angl. *Area Under Curve*, kur *Curve* yra ROC kreivė) [4]. Kuo kreivė arčiau 1 visuose laiko taškuose, tuo modelis geresnis (1 pav.). Išvados dėl tinkamiausio modelio panašios kaip ir naudojant konkordancijos koeficientą. Veibulo AFT bei Kokso PH modeliai pasirodo labai panašiai, o RSF beveik visoje laiko skalėje pranašesnis.



1 pav. Dinaminis AUC apmokymo aibėje.

5 Išvados

Pritaikius tris išgyvenamumo analizės metodus galima teigti, jog toks požiūris personalo kaitai analizuoti yra tinkamas. Atsitiktiniai išgyvenimo miškai lenkia Kokso PH ir Veibulo AFT modelius pagal konkordancijos indeksą. Tik RSF modelis apmokymo aibėje fiksuoja rezultatą, aukštesnį nei 0,8 ir vienintelis testavimo aibėje viršija 0,6. Iš dinaminio AUC išvados tokios pat – RSF beveik visame laiko intervale pranašesnis. Papildomas šio modelio privalumas tas, kad jį ir pritaikyti yra lengviausia.

Literatūra

[1] Alamsyah, A., Salma, N. *A Comparative Study of Employee Churn Prediction Model*. 4th International Conference on Science and Technology (ICST), 2018, pp. 1-4.

[2] Collett, D. *Modelling Survival Data in Medical Research*. British Actuarial Journal, 1995, **1**(2).

[3] Jin, Z., Shang, J., Zhu, Q., Ling, C., Xie, W., Qiang, B. *RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis*. Web Information Systems Engineering – WISE 2020. Lecture Notes in Computer Science, 2020, **12343**.

[4] Kamarudin, A.N., Cox, T., Kolamunnage-Dona, R. *Time-dependent ROC curve analysis in medical research: current methods and applications.* BMC Med Res Methodol, 2017, **17**, 53.

[5] Kleinbaum, D., Klein, M. *The Cox Proportional Hazards Model and Its Characteristics.* Survival Analysis: A Self-Learning Text, 2005, 2, pp. 97-159.

[6] Kleinbaum, D., Klein, M. *Parametric Survival Models.* Survival Analysis: A Self-Learning Text, 2005, 2, pp. 289-361.

[7] Morita, J. G., Lee, T. W., Mowday, R. T. *The regression-analog to survival analysis: A selected application to turnover research*. Academy of Management Journal, 1993, **36**(6), pp. 1430-1464.

Vairavimo maršruto skaičiavimo, grindžiamo skatinamuoju mokymusi, vizualios aplinkos kūrimas

Oskaras Klimašauskas, Gintautas Dzemyda

Vilniaus universitetas, Duomenų mokslo ir skaitmeninių technologijų institutas,
Akademijos g. 4, LT-08412 Vilnius,
oskaras.klimasauskas@mif.vu.lt

Santrauka. Straipsnyje yra sprendžiamas optimalaus maršruto kelių tinkle paieškos uždavinys. Uždavinys yra modelinis, nes kelių tinklas pasirinktas stačiakampis su vienodomis tiesiomis atkarpomis, o kai kuriose sankryžose yra veikiantis šviesoforas. Uždavinys sprendžiamas naudojantis skatinamojo mokymosi algoritmais. Straipsnyje siekiama palyginti skirtingus skatinamojo mokymosi algoritmus, o taip pat sukurti vizualią aplinką, leidžiančią stebėti skatinamojo mokymosi procesą. Vizuali aplinka yra sudaryta iš automobilio, kelių ir šviesoforų tinklo, bei galutinio finišo. Mokymasis vyksta siekiant minimizuoti pravažiuotų atkarpų skaičių. Algoritmai, sunaudojantys mažiausią tokių atliktų žingsnių skaičių ir tuo būdu randantys sprendimą greičiausiai, yra geriausi. Tyrime buvo naudojami keturi skatinamojo mokymosi algoritmai: *Q-learning, Sarsa, Sarsa(λ), Actor-critic*. Pasiūlytos realizacijos, labiausiai tinkančios sprendžiamam uždaviniui. Aplinka naudinga susipažįstantiems su skatinamuoju mokymusi ir jo principais. Straipsnyje pateikiama nuoroda į aplinkos programos kodą ir instrukcijos, kaip ja pasinaudoti. Tai turėtų išplėsti skatinamojo mokymosi taikymus.

Raktiniai žodžiai: Mašininis mokymasis, Skatinamasis mokymasis, Maršruto paieška, Demonstracinė aplinka.

1 Įvadas

Skatinimasis mokymasis yra mašininio mokymo atšaka, kur egzistuoja tam tikras agentas, kuris mokosi spręsti sudėtingą uždavinį, atlikdamas elementarius veiksmus, bet siekdamas maksimizuoti kažkokį ilgalaikį atlygį ar pasiekimą. Skatinamasis mokymasis pagrįstas agento ir aplinkos sąveika: agentas stebi aplinką, pagal ją atlieka veiksmą, gauna atlygį priklausomai nuo jo veiksmų. Šių atlygių pagrindu agentas tobulina savo veiksmų strategiją.

DOI: https://doi.org/10.15388/LMITT.2024.7

Pagrindinis tikslas yra išmokti optimalią veiksmų seką, kuri leistų pasiekti geriausią galimą rezultatą konkrečioje užduotyje. Matematiškai apskaičiuoti geriausią rezultatą dažnai arba labai sunku, arba užtruktų labai daug laiko, arba visiškai neįmanoma. Skatinamasis mokymasis yra vienas iš būdų, kad surasti apytiksliai optimalią veiksmų seką.

Šiame straipsnyje nagrinėjamas uždavinys susijęs su optimalaus maršruto radimu modelinėje aplinkoje, kurioje kelių tinklas yra pavaizduotas kaip stačiakampė tinklas, sudarytas iš vienodų atstumų segmentų ir įrengtų šviesoforų tam tikrose sankryžose. Siekiant išspręsti šį uždavinį, straipsnyje taikomi ir analizuojami įvairūs skatinamojo mokymosi algoritmai: *Q-learning* [1]*, Sarsa* [2]*, Sarsa(λ)* [3] ir *Actor-critic* [4]. Šie metodai pasirinkti todėl, kad jų veikimas grindžiamas skirtingais principais arba bendrumu. Šių metodų efektyvumas lyginamas, kad būtų nustatytas efektyviausias algoritmas optimalaus maršruto paieškai. Tyrimas leidžia ne tik identifikuoti geriausias praktikas optimaliems maršrutams rasti, bet ir suteikia galimybę giliau suprasti, kaip skirtingi skatinamojo mokymosi algoritmai veikia.

Darbo tikslas yra palyginti skirtingus skatinamojo mokymosi algoritmus, o taip pat sukurti vizualią aplinką, leidžiančią stebėti skatinamojo mokymosi procesą. Tokia stebėjimo galimybė yra geras būdas pažinti skatinamąjį mokymąsi, jo veikimą.

2 Sprendžiamo uždavinio formuluotė

Uždavinys sudarytas iš prieš tai minėto kelių tinklo bei šviesoforų, kurie turi 2 stadijas: žalia ir raudwona. Uždavinys turi *n* įvažiavimų į kelių tinklą, į vieną kurių agentas (mūsų atveju tai vairuotojas) atsitiktinai įvažiuoja, ir *m* išvažiavimų, į bet kurį patekti yra to agento tikslas. Vairuotojas turi penkis judesių pasirinkimus: judėti į kairę, dešinę, žemyn, aukštyn, nejudėti. Jo galimi judesiai yra tose kryptyse, kuriose egzistuoja kelias ir jame nėra degančio raudono šviesoforo, nejudėti gali pasirinkti bet kokioj situacijoj.

Įsiveskime laiko sąvoką. Mokymosi laiką žymėsime *t*. Pradiniu momentu $t = 0$. Vienas laiko žingsnis yra sugaištamas kiekvienam agento pasirinktam veiksmui, t. y. po atlikto veiksmo *t* padidėja vienetu. Šviesoforų spalvos keičiasi (pasidaro iš žalios į raudoną, ar iš raudonos į žalią) sinchroniškai, kas *k* laiko žingsnių, t. y. kas *k* agento pasirinktų judesių.

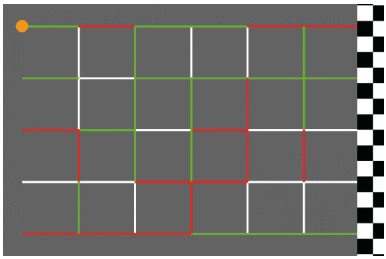
Agento dabartinė būsena yra jo koordinatės ir šviesoforų stadija. Kadangi šviesoforai keičiasi visi tuo pačiu metu, jų stadija globaliai yra reprezen-

tuojama kaip 1 arba -1, kur stadijos reikšmė pasikeičia kas *k* laiko žingsnių. Agento būseną momentu *t* galime aprašyti kaip (*x*, *y*, šviesoforų stadija), čia (*x, y*) yra sankryžos, kurioje randasi automobilis, koordinatės.

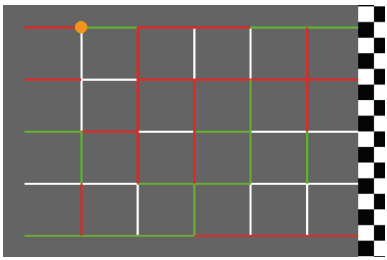
Skatinamajame mokymesi agentas gauna atlygį už sėkmę, ir yra baudžiamas už nesėkmę. Mūsų atveju agentas gauna +1 atlygį už pasiektą finišą ir -1 atlygį už paeitą žingsnį – kuo ilgiau agentas eis iki finišo, tuo labiau jis bus nubaustas. Siekiant gauti kuo didesnį atlygį, optimalus maršrutas turės mažiausiai panaudotą žingsnių skaičių. Žinoma, mūsų atveju visada atlygis bus neigiamas.

3 Aplinkos aprašymas

Vizuali aplinka yra sudaryta iš automobilio, kelių ir šviesoforų tinklo, bei galutinio finišo. Mokymasis vyksta siekiant minimizuoti pravažiuotų atkarpų skaičių. Numatyta galimybė pasirinkti skirtingus skatinamojo mokymosi algoritmus šiam uždaviniui spręsti. Algoritmai, sunaudojantys mažiausią tokių atliktų žingsnių skaičių ir tuo būdu randantys sprendimą greičiausiai, yra geriausi. Tyrime buvo naudojami keturi skatinamojo mokymosi algoritmai: *Q-learning, Sarsa, Sarsa(λ), Actor-critic*. Sukurta aplinka 1 pav., iliustruojanti skatinamojo mokymosi procesą, yra tinklas kelių ir šviesoforų, kuriame yra penkios pradinės (starto) būsenos kairėje, kurios atsitiktinai parenkamos agentui momentu $t = 0$, ir penkios finišo būsenos dešinėje, vieną kurių pasiekus pasibaigia simuliacija. Šviesoforų išsidėstymas yra atsitiktinai sugeneruojamas ant kelių tinklo. Šviesoforų atskirai nepaišysime. Sankryža, kur galimo judesio atkarpa yra nuspalvinta baltai reškia, kad ta kryptimi judėjimas nėra reguliuojamas šviesoforu ir ten galima pravažiuoti bet kokiu atveju. Ten kur žalia arba raudona, reiškia kad yra šviesoforas, kuris persijungia kas $k=1$ laiko momentų. Pradinės būsenos irgi suprantamos kaip sankryžos su galimu judėjimu tik į priekį ir tos sankryžos irgi gali būti reguliuojamos šviesoforu. Sankryžoje galima judėti ten, kur „dega“ žalia šviesa arba kur nėra eismo reguliavimo šviesoforu. Agentas turi penkis judesių pasirinkimus: judėti į kairę, dešinę, žemyn, aukštyn, nejudėti. Agento galimi judesiai yra ten, kur egzistuoja kelias ir jame nėra degančio raudono šviesoforo, o nejudėti gali pasirinkti bet kokioj situacijoj. 2 pav. atveju galimi judesiai šiuo atveju yra judėti į dešinę arba palaukti bent vieną laiko žingsnį vietoje. 1 pav. atveju jo galimi judesiai yra į dešinę, žemyn, nejudėti. Veiksmą jis pasirenka pagal naudojamo algoritmo apibrėžtas taisykles. Šviesoforų stadija keičiasi (pasidaro iš žalios į raudona, ar iš raudonos į žalią) kas kiekvieną laiko žings-



1 pav. Aplinkos stadija antrame laiko žingsyje



2 pav. Aplinkos pradinė stadija

nį. Šiuo atveju visi šviesoforai persijungs iš vienos stadijos į kitą priklausomai nuo *t* tokiu būdu: -1^t, kur *t* yra dabartinis laiko žingsnis.

4 Pritaikyti skatinamojo mokymosi algoritmai

Šiame straipsnyje keturi skirtingi skatinamojo mokymosi algoritmai buvo naudojami geriausiam maršrutui surasti. Agentas yra baudžiamas už kiekvieną praeitą žingsnį – kuo ilgiau eina, tuo daugiau prisirenka baudų. Agento tikslas tampa pasiekti bet kurį išėjimo tašką per kuo mažiau žingsnių.

Tyrinėjami buvo *off-policy, on-policy, eligibilty trace, policy gradient* tipo algoritmai:

1. *Q-learning (off-policy)*
2. *Sarsa (on-policy)*
3. *Sarsa(λ) (eligibility trace)*
4. *Actor-critic (policy gradient)*

Bendru atveju skatinamojo mokymosi algoritmuose disponuojama su [5]:

- Būsenų rinkiniu. Būsenos apima visas galimas agento vietas aplinkoje.
- Veiksmų rinkiniu. Veiksmai apima visus galimus veiksmus, kuriuos agentas gali paimti aplinkoje.
- Apdovanojimais/baudomis. Atlygio vertė priklauso nuo agento būsenos.
- Nuolaidos faktoriumi. Nuolaida nusako kiek vertinam ilgalaikius atlygius lyginant su trumpalaikiais.
- Elgesiu (*Policy*). Elgesys nusakomas veiksmų pasirinkimo taisykle, kuri apibrėžia tikimybes, su kuriomis agentas pasirenka kiekvieną galimą veiksmą.

Q-learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma\, Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (1)$$

Q-learning yra *off-policy* metodas, jis sukuria sąryšių lentelę *Q* tarp visų būsenų *S* ir veiksmų *A* porų, vadinamą *Q-table*. Reikšmė lentelėje yra numatomas atlygis pasirenkant konkretų veiksmą esamoje būsenoje, kur $Q(S_t, A_t)$ yra dabartinio laiko žingsnio *t* ir jame pasirinkto veiksmo sąryšio reikšmė. *Q-learning* optimizuoja *off-policy control*, tai reiškia, kad jis būsenoj S_t pasirenka veiksmą A_t pagal veiksmų pasirinkimo taisyklę (*behaviour policy*), tačiau atnaujina matricos *Q* reikšmes pagal tikslo siekimo taisyklę (*target policy*). Šiuo atveju tikslo siekimo taisyklė (*target policy*) yra godi (*greedy*), kur pasirenka veiksmą, turintį didžiausią reikšmę būsenoje, o veiksmų pasirinkimo taisyklė (*behaviour policy*) yra *ε*-godi (*ε-greedy*), kur renkasi veiksmą su didžiausia reikšme, bet turi tikimybę *ε* atsitiktinai pasirinkti bet kokį veiksmą (iš galimų veiksmų toje būsenoje). Atnaujinimo formulėje (1) dabartinės būsenos S_t ir veiksmo A_t poros reikšmė $Q(S_t, A_t)$ yra atnaujinama remiantis sekančios būsenos (būsena, kurioje atsirandame atlikus veiksmą A_t) ir godaus (*greedy*) veiksmo pora $\max_a Q(S_{t+1}, a)$, γ nurodo nuolaidos faktorių, t.y. kiek norime atsižvelgti į sekančių žingsnių reikšmes. R_{t+1} nusako būsenos po veiksmo A_t atlygį, *α* yra žingsnio dydis dar kitaip vadinamas mokymosi greičiu.

Sarsa

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma\, Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \quad (2)$$

Sarsa, kitaip nei *Q-learning* yra *on-policy* metodas. Jis taip pat atnaujina tas pačias būsenų *S* ir veiksmų *A* poras $Q(S_t, A_t)$ reikšmes, tačiau tai daro optimizuojant agento veiksmų pasirinkimo taisyklę (*behaviour policy*). Vietoj to, kad atnaujintų *Q* reikšmes pagal godžiai (*greedy*) pasirinktą veiksmą, jas atnaujina pagal sekantį veiksmą A_{t+1} (2), pasirinktą *ε*-godžiai (*ε-greedy*). Kol *Q-learning* mokinasi idealius veiksmus nepaisant to, kad gali atsitiktinai pasirinkti kitą, *Sarsa* tiesiogiai mokinasi pagal tuos veiksmus kuriuos pasirenka.

Sarsa(λ)

Sarsa(λ) yra *Sarsa* algoritmo išplėtimas, kuriame naudojamas parametras λ ($0 \leq \lambda \leq 1$), leidžiantis kontroliuoti, kiek ankstesnės patirtys turi įtakos būsenos ir veiksmo poros reikšmės atnaujinimui. Algoritmas naudoja žymeklių

masyvą (*eligibility traces*) z_t, kuriame kaupiama, kaip seniai viena ar kita būsena buvo aplankyta. Žymeklio reikšmė nustatoma 1 naujausioje būsenoje, ir sumažėja parametro λ dydžiu kas žingsnį, tokiu būdu agentas gauna žymeklių „uodegas" (*traces*), kur ankstesnės patirties reikšmė mažėja, kuo ilgiau būsena neaplankyta. Kai $\lambda = 0$, *Sarsa(λ)* tampa paprastu *Sarsa* algoritmu, kuris žiūri tik į sekančio veiksmo numatomą atlygį, o jei 1, tada tampa „*Monte Carlo*" metodu [6]. *Q* matricos elementų (įtakos būsenos ir veiksmo poros reikšmės) perskaičiavimas vykdomas naudojant *Q* matricos aproksimaciją [6], [7], kur esant didelei aplinkai, jos būsenai reprezentuoti naudojamas tam tikras ypatybių vektorius. Kai aplinka yra pakankamai maža, kad galima į būsenų ir veiksmų porų reikšmes žiūrėti individualiai, tokiu atveju siūlome z_t naudoti kaip lentelę, kur kaupiamos visos žymeklių reikšmės (3). Dabartinei būsenai ir veiksmui $z(S_t, A_t)$ priskiriama reikšmė 1, o visos kitos reikšmės pamažėja λ dydžiu. Čia matrica Q_t yra visos matricos *Q* reikšmės *t* laiko momentu. Algoritmas veiksmus taip pat rinkosi ε-godžiai (*ε-greedy policy*).

$$\begin{aligned} &z_t \leftarrow \gamma\lambda z_{t-1} \\ &z(S_t, A_t) \leftarrow 1 \\ &Q_t \leftarrow Q_t + \alpha\big[(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))\big]z_t \end{aligned} \tag{3}$$

Mūsų tyrime algoritmai *Q-learning*, *Sarsa*, *Sarsa(λ)* naudojo tokius parametrus: mokymosi žingsnio dydį $\alpha = 0.01$; nuolaidos faktorių $\gamma = 1$, kad pilnai atsižvelgtų į ateities atlygius; aplinkos tyrinėjimo vertę $\varepsilon = 0{,}1$; pradinės *Q* vertes $Q(s, a) = -100$, visiems $s \in S$, $a \in A(s)$. Šios pradinės *Q* reikšmės buvo pasirinktos todėl, kad aplinkoje žingsnių atlygis yra neigiamas. Jei $Q(s, a)$ būtų inicijuota nulinė, tada agentui ilgą laiką aukščiausios vertės veiksmas būtų mažiausiai tyrinėtas, nors ir blogas, veiksmas. *Sarsa(λ)* taip pat naudojo $\lambda = 0{,}95$ parametrą žymekliams, nes tai rodo, kad ankstesnės patirtys turi ganėtinai didelę įtaką, bet vis tiek pamažu į jas mažiau atsižvelgiama.

Actor-critic

Actor-critic yra *policy* gradient metodas. Vietoj to, kad išmoktų veiksmų vertes ir tada pagal kažkokią veiksmų pasirinkimo taisyklę juos pasirinktų, jis tiesiogiai išmoksta elgseną gradiento metodu. *Actor-critic* užtat skiriasi nuo paprasto *policy* gradient tuo, kad jis taip pat mokosi būsenos vertę $v(S_t)$, kuri nusako, kaip gerai agentui būti tam tikroje būsenoje, tikslu kritiškai vertinti savo elgsenos atnaujinimą. Kaip ir su *Sarsa(λ),* esant didelei aplinkai Q

matricos elementų (įtakos būsenos ir veiksmo poros reikšmės) perskaičiavimas vykdomas naudojant būsenos vertė $\hat{v}$ aproksimacija [4]. Nedidelės aplinkos atvejui siūlome visas būsenų vertes $v(S_t)$ kaupti atskirai (4). Tam, kad algoritmas įgautų *policy* π (veiksmų tikimybių paskirstymą) naudojama *softmax* funkcija. Funkcijai duodama reikšmė yra *policy* parametras θ, kuris įvertina kiek stipriai reikia apsvarstyti veiksmą. Šiuo atveju naudojam skirtingus žingsnio dydžius α būsenai α^v ir *policy* parametrui α^θ. Siekiant greičiau išmokti apytikslią būsenos vertę ir pagal ją geriau kritiškai vertinti savo elgseną, būsenos mokymosi greitį nustatome didesniu. $v(S_{t+1})$ (arba $\hat{v}(S_{t+1}, w)$) yra sekančios būsenos reikšmė.

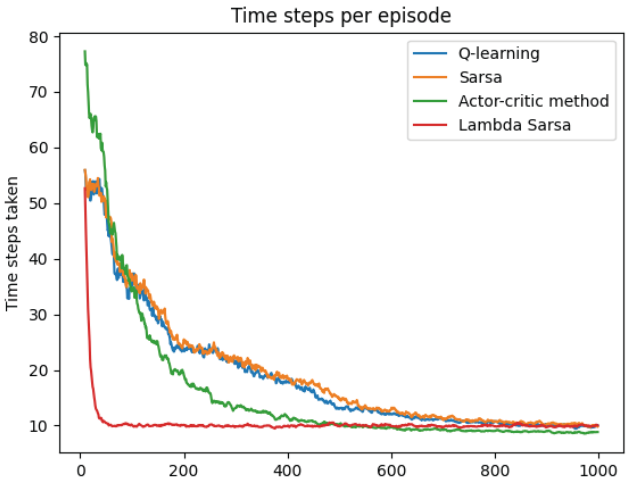
$$\begin{aligned} v(S_t) &\leftarrow v(S_t) + \alpha^v[R_{t+1} + \gamma v(S_{t+1}) - v(S_t)] \\ \theta &\leftarrow \theta + \alpha^\theta[R_{t+1} + \gamma v(S_{t+1}) - v(S_t)]\nabla ln\pi(A|S,\theta) \end{aligned} \tag{4}$$

Actor-critic skyrėsi nuo kitų trijų aukščiau pateiktų algoritmų tuo, kad naudojo papildomai $\alpha^v = 0{,}1$ ir $\alpha^\theta = 0{,}005$, kad greičiau galėtų tiksliau kritiškai vertinti savo veiksmus. Vietoj būsenų ir veiksmų porų jis naudojo tik numatomas būsenų vertes, kurios inicializuojamos $v(s) = 0$, visiems $s \in S$. Šios pradinės vertės yra standartinės ir nėra labai svarbios. *Policy* parametrai θ buvo atsitiktinai sugeneruojami intervale (0, 1) visiems $s \in S$, $a \in A(s)$ kad pradžioj turėtų ne vienodą veiksmų pasirinkimą.

Tyrime lyginame *Q-learning* mokymąsi pagal atskirą tikslo siekimo taisyklę (*target policy*), *Actor-critic* tiesioginį elgsenos mokymąsi bei skirtumą tarp *Sarsa* su žymekliais ir be jų.

5 Tyrimo rezultatai

Eksperimentiškai ištirti keturi aukščiau pateikti algoritmai. 3 pav. pateikti duomenys yra 20 mokymosi paleidimų vidurkis, rodantis reikalingų žingsnių skaičių finišui pasiekti, didėjant mokymosi epochų kiekiui. Kaip matome, *Sarsa(λ)* greičiausiai randa optimalų 8 žingsnių kelią ir jį pasiekia apie 60-oj epochoj. Tuo tarpu *Actor-critic* algoritmui prireikia 500 epochų, o *Q-learning* ir paprastam *Sarsa* algoritmui prireikia 800 epochų. Kadangi tiek *Sarsa* algoritmai, tiek *Q-learning* naudoja ε-godų veiksmų pasirinkimą, net ir išmokę optimalų kelią, jie kartais atsitiktinai pasirenka neteisingą veiksmą. Priešingai, *Actor-critic* metodas atnaujina savo elgseną ir palaipsniui mažina aplinkos tyrinėjimą. Tai galima pastebėti nuo 600-osios epochos, kai *Actor-critic* pradeda rečiau rinktis atsitiktinius veiksmus ir artėja prie 8 žingsnių



3 pav. Algoritmų palyginimo rezultatai

vidurkio. Paprastas *Sarsa* ir *Q-learning* šiame uždavinyje parodė identiškus rezultatus, reikšmių optimizavimas pagal atskirą elgseną neturėjo didelės įtakos. *Actor-critic* pradeda lėčiausiai, tačiau po 100 epochų aplenkia *Q-learning* ir *Sarsa* ir po 600 epochų aplenkė *Sarsa(λ)*. Tiesiogiai mokydamasis elgseną, šis metodas pasižymi tuo, kad išmoksta lygiaverčius veiksmus ir gali juos pasirinkti su lygia tikimybe. Šioje aplinkoje tai reiškia, kad jis išmoksta, jog tam tikroje situacijoje tiek pat verta važiuoti į viršų, kiek ir į apačią.

6 Išvados

Tyrime buvo naudojami keturi skatinamojo mokymosi algoritmai: *Q-learning*, *Sarsa*, *Sarsa(λ)*, *Actor-critic*. Pasiūlytos realizacijos, labiausiai tinkančios sprendžiamam uždaviniui. Remiantis tyrimo rezultatais, galima teigti, kad

1. *Sarsa(λ)* algoritmas buvo efektyviausias randant optimalią maršrutą greitai. Naudodamas ankstesnius savo patyrimus būsenos ir veiksmų poros reikšmių atnaujinimui, jis sugeba daugiau nei aštuonis kartus greičiau išmokti optimalų kelią.
2. *Actor-critic* metodas, nors ir pradeda mokintis lėčiausiai, pasižymėjo tuo, kad gali išmokti lygiaverčius veiksmus bei sumažina būsenų tyrinėjimą laikui einant.

3. *Q-learning* ir paprastas *Sarsa* reikalauja mažiau skaičiavimų ir juos lengviau įgyvendinti, tačiau jie nepasirodo tiek įspūdingai kiek *Sarsa*, jei *Sarsa* naudoja *eligibilty traces*.

Sukurta vizuali aplinka yra sudaryta iš automobilio, kelių ir šviesoforų tinklo, bei galutinio finišo. Ji leidžia stebėti agento mokymosi procesą sulėtintame režime. Mokymasis vyksta siekiant minimizuoti pravažiuotų atkarpų skaičių. Numatyta galimybė pasirinkti skirtingus skatinamojo mokymosi algoritmus šiam uždaviniui spręsti. Aplinka naudinga susipažįstantiems su skatinamuoju mokymusi ir jo principais. Straipsnyje pateikiama nuoroda į aplinkos programos kodą ir instrukcijos, kaip ja pasinaudoti. Tai turėtų išplėsti skatinamojo mokymosi taikymus. Aplinką galima išbandyti ir stebėti, kaip vyksta skatinamojo mokymosi procesas naudojantis šia nuoroda Nacionaliniame atviros prieigos mokslinių tyrimų duomenų archyve MIDAS [8].

Literatūra

[1] C. Watkins, P. Dayan, Technical Note: Q-Learning, *Machine Learning*, 8, 279-292., 1992.
[2] G. A. Rummery, M. Niranjan, *Online Q-Learning using Connectionist Systems*, Technical Report CUED/F-INFENG/TR 166, 1994.
[3] R. S. Sutton, H. V. Seijen, M. C. Machado, P. M. Pilarski, A. R. Mahmood, *True Online Temporal-Difference Learning*, Journal of Machine Learning Research (JMLR), 17(145):1-40, 2016.
[4] R. S. Sutton, M. White, T. Degris, *Off-Policy Actor-Critic*, https://doi.org/10.48550/arXiv.1205.4839, 2012.
[5] A. Daranda, G. Dzemyda, Reinforcement learning strategies for vessel navigation *Integrated Computer-Aided Engineering*, 30 (1), 53-66, 2023.
[6] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction. Second edition, MIT Press, 281, 303-305, 2018.
[7] H. v. Hasselt, DeepMind x UCL RL Lecture Series, 2021. https://www.youtube.com/watch?v=TCCjZe0y4Qc&list=PLqYmG7hTraZDVH599EItlEWsUOsJbAodm
[8] O. Klimašauskas, G. Dzemyda, „Vairavimo maršruto skaičiavimo, grindžiamo skatinamuoju mokymusi, vizualios aplinkos kūrimas," Nacionalinis atviros prieigos mokslinių tyrimų duomenų archyvas (MIDAS), 2024. https://www.doi.org/10.18279/MIDAS.RLmokymasis.250217

Semantic Segmentation for Change Detection in Satellite Imaging

Kürşat Kömürcü, Linas Petkevicius

Vilnius University, Institute of Computer Science,
Didlaukio str. 47, LT-08303 Vilnius, Lithuania
kursat.komurcu@mif.stud.vu.lt, linas.petkevicius@mif.vu.lt

Abstract. Change detection is a common and actual problem in the field of remote sensing. The classical approaches using raw pixel information are very sensitive to noise. In this study we propose the usage of additional semantic information for change detection. We use the semantic segmentation methods like geospatial Segment Anything Model and encoder based U-Net to evaluate the predictions and tracing the semantic information as well as raw information in change detection. Later the multidimensional time series data is used via the Vector Autoregression model to predict the future changes in the landscape. The observations which fall out of the prediction interval are considered as the changes in the landscape. The proposed method is evaluated on the dataset of the random locations across the Baltic region. The research is accompanied by the data and reproducible code at Github repository[1].

Keywords: Deep learning, semantic segmentation, change detection, satellite imagery, Vector Autoregression

1 Introduction

Change detection is a problem to object variations by observing them over time. Satellite imagery is one of the domains where change detection is widely used. Remote sensing change detection is applied by using data from Earth-orbiting satellites and identifying the changes in the landscape structural permutations which have happened throughout a specific time track.

The change detection problem is analysed in classical mathematical modelling methods as algebraic analysis difference [1] or regression [2]. However, any methods based only on pixel intensity are very sensitive

[1] https://github.com/kursatkomurcu/semantic_segmentation_for-_change_detection_in_satellite_imaging

DOI: https://doi.org/10.15388/LMITT.2024.8

to noise. This common problem in computer vision, thus deep learning methods could be used to overcome this problem [3]. Deep learning methods for change detection could be divided into two groups based on different approaches. First approach uses 3-channel RGB images, while the second approach is based on multi-spectral imagery and contains more band information.

The RGB imagery approach is widely used in the field of remote sensing. The most popular method is based on U-Net [4] architecture. The U-Net architecture is used in the field of remote sensing for semantic segmentation [5] and change detection [6]. The U-Net architecture is also used in the field of remote sensing for change detection in combination with Siamese networks [7] demonstrating good results on cases like OSCD dataset [8]. Some recent methods apply deep learning models using bi-temporal images [9] or transformer based models [10].

The main challenges arise among approaches. The older satellite images do not enable them to form bi-temporal images. The bi-temporal images are formed by using the same place image taken at different times. The relief displacement is also a problem in remote sensing [6]. The relief displacement is a problem that occurs when the same object is imaged from different angles. Finally the seasonality of weather conditions is also a problem in remote sensing when climate conditions are changing, thus the same place could be imaged in different seasonality and weather conditions.

The addressing of challenges in remote sensing the deep learning methods are suitable choices. While some successful feature extraction methods could be used using end-to-end models [11] it raises the computational challenges. To address this we focus on the semantic segmentation methods. Very common approach is the usage of U-Net models modifications for semantic segmentation [12]. On the other hand, a recent successful model for semantic segmentation is Segment Anything Model [13] which was adopted to remote sensing imaging [14].

The most common non-commercial application of change detection falls under climate monitoring. For this current resolution of satellite imagery is rather sufficient. The 10 to 60 meter resolution is from Sentinel-2 [15]. The land site change detection for coastal zones is analysed [16].

The change detection is significantly impacted on noises and quality of images [8] or non stationary objects in the images like [17]. Thus it

encounters the problems of joining datasets [18]. The class imbalance is also the common challenge [19, 20, 17, 21], mostly since background class in general is not changing [18]. Finally, clouds and noises are also a challenge [22] as increased saturation [17] or distorted colours [23].

The work is organised as follows. In Section 2, we discuss the proposed methodology. In Section 3, we discuss the dataset. In Section 4, we discuss the results. In Section 5, we discuss the conclusions.

2 Methodology

2.1 Semantic Segmentation

Semantic segmentation is a computer vision problem of assigning a class label to each pixel in an image from a predefined set of classes. Let's assume the input of format $X \in R^{c \times w \times h}$ of image consistent of tensor X with c - number channels, and width/height w, h respectively. The semantic segmentation mask $X \in R^{c \times w \times h}$ contain L number of classes, where each pixel is assigned to one of the classes. Such models could predict the class of each pixel in the image [12]. In our experiments we used the U-Net like model[2]. The pre-trained model had Building, Land, Road, Vegetation, Water and Unlabeled classes. For the generic segmentation models like Segment Anything Model [13] provide object mask prediction confidence score.

Upon completion of the segmentation process, the class probabilities for each pixel are aggregated to calculate average class probabilities for each image, forming a summarised representation of the segmentation outputs. To integrate these segmentation results into the VAR model, we construct feature vectors for each temporal pair of images by computing differences in the aggregated class probabilities between the two time points, thus capturing the changes in class distributions over time. These feature vectors are then weighted by the confidence scores derived from the segmentation phase, ensuring that the VAR model input emphasises data with higher predictive reliability.

2.2 Vector Autoregression

The vector autoregression (VAR) is a model used to capture the linear interdependencies among multiple time series data. The VAR model is a

[2] https://github.com/ayushdabra/dubai-satellite-imagery-segmentation

generalisation of the univariate autoregressive model (AR) [24]. The VAR model is used to forecast tasks. The VAR model is defined as follows:

$$y_t = \beta + \sum_{i=1}^{p} \Omega_t Y_{t-i} + \epsilon_t$$

where y_t is a $k \times 1$ vector of endogenous variables at time t, β is a $k \times 1$ vector of bias, Ω is a $k \times k$ matrix of coefficients for i-th lag, p is the order of the VAR process, and ϵ_t is a $k \times 1$ vector of error terms at time t. The confidence interval of VAR models could be used either dynamic, or fixed. In our case use t distribution confidence interval which is the same for each time step. The critical value for the confidence level α, in our experiments we used α = 0.05. The VAR model was used using *Python* package *statsmodel*.

By transforming the class probabilities and confidence scores into a time-series format, we enable the VAR model to utilise these inputs effectively for forecasting landscape changes, ensuring that the transition from image data to predictive modelling is both smooth and logical.

3 Dataset

The investigation of change detections covered a wide range of diverse cases. We randomly chose 100 coordinates over the Baltic region (53,53100 - 59,69747 latitude values and 20,49722 - 28,22760 longitude) using uniform distribution. After that, we used COPERNICUS/S2 satellite in Google Earth Engine API to collect images of random chosen coordinates over the 2022 - 2023 time period. In our experiments, we used pixel intensities of B4, B3 and B2 bands which represent red, green and blue colours. For each coordinate, we made predictions using *geospatial* Segment Anything Model [12] and collected IOU and score values. Class probabilities are collected using the U-Net model. Cloud Probabilities collected using Google Earth Engine API. In such the dataset for each coordinate consist of 11 features of Raw Pixel Intensity of B4, Pixel Intensity of B3, Pixel Intensity of B2, IOU, Scores, Probabilities of 6 classes and Cloud Probabilities.

Table 1. Summary table

Index	Lat	Lon	RMSE	AIC	Fall In CI
0	57.9822	27.5759	0.085	-92.993	%0
1	54.8303	21.8945	0.118	-123.880	%0
2	59.1785	24.5851	0.058	-92.618	%0
3	57.2123	24.1739	<0.001	None	%99.363
4	55.4973	23.1317	0.089	-105.224	%0
5	59.5876	25.7885	0.053	-96.585	%0.746
6	57.2948	22.5929	0.093	-98.745	%0
7	53.6124	27.2380	0.080	-102.598	%0
8	54.6356	22.8023	0.096	-113.109	%0
9	56.6370	20.7791	0.112	-123.783	%0

Note: RMSE and Fall In CI columns are values for Class2

4 Results

For each point, we collected sentinel-2 RGB images using scale 10 zoom rate. Then, having the surrounding environment around the segmentation predictions was made using relevant models for each image. Such enables semantic information for each investigative pixel. After creating our dataset, we used VAR model for selected index and forecast h = 12 steps. The experiment we calculated root mean square error (RMSE), akaike information criterion (AIC) and confidence intervals for each feature using t distribution.

Figure 1 presents the general pipeline of approaches. The segmented image semantic information is added to vector time series models, thus while raw image data seems unchanged significantly, the semantic information allows an additional control mechanism for quality assessment. Cloud probability is often used to remove untruthful images, the same could be done by tracking unchanged situations. The illustrative case in Figure 1 can be seen for index 3 in the Table 1 above. Also one can see in Table 1 that some testing images have high variation in raw data or some data was not overlapped (black/empty image) over specific flight and 0 observations fell in the confidence interval.

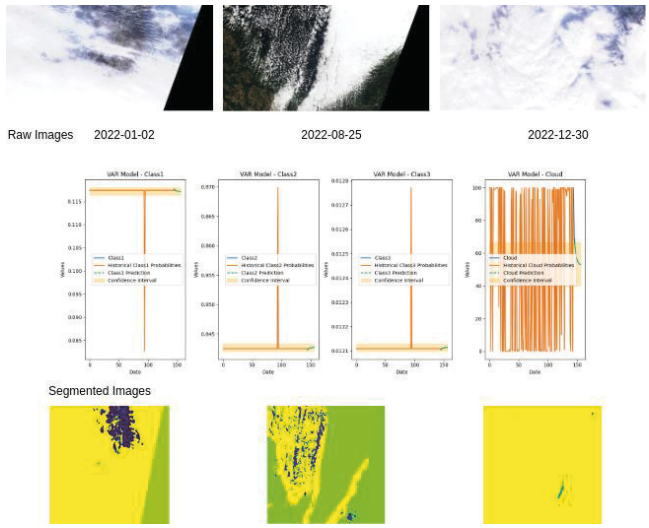


Figure 1. The illustrative example of confidence interval of prediction of the VAR model, which is used to detect the changes in the landscape.

5 Conclusions

In the study we investigate the estimation of non-changing temporal situations in satellite imagery. The publication proposes the addition of additional semantic information usage for tracking changes. The raw and semantic information modelled by vector auto-regressive models. The experiments demonstrated successful usage of the method. The identified change-detection cases are often related to data obscures of not visible areas. The publication is complemented with a reproducible repository of method pipeline in Github[3].

References

[1] Ke, L., Lin, Y., Zeng, Z., Zhang, L., & Meng, L. (2018). Adaptive change detection with significance test. IEEE Access, 6, 27442-27450.

[2] Ridd, M. K., & Liu, J. (1998). A comparison of four algorithms for change detection in an urban environment. Remote Sensing of Environment, 63(2), 95-100.

[3] https://github.com/kursatkomurcu/semantic_segmentation_for-_change_detection_in_satellite_imaging

[3] Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1-2), 305-307.
[4] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany.
[5] Peng, D., Zhang, Y., & Guan, H. (2019). End-to-end change detection for high-resolution satellite images using improved unet++. Remote Sensing, 11(11), 1382.
[6] Gong, J., Hu, X., Pang, S., & Li, K. (2019). Patch matching and dense crf-based co-refinement for building change detection from bi-temporal aerial images. Sensors, 19(7).
[7] Daudt, R. C., Le Saux, B., & Boulch, A. (2018). Fully convolutional siamese networks for change detection. Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece.
[8] Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2018). Urban change detection for multispectral earth observation using convolutional neural networks. IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain.
[9] Zheng, Z., Ma, A., Zhang, L., & Zhong, Y. (2021). Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery. Proceedings of the IEEE/CVF International Conference on Computer Vision.
[10] Zhang, C., Wang, L., Cheng, S., & Li, Y. (2022). Swinsunet: Pure transformer network for remote sensing image change detection. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-13.
[11] Liu, J., Xuan, W., Gan, Y., Zhan, Y., Liu, J., & Du, B. (2022). An end-to-end supervised domain adaptation framework for cross-domain change detection. December 2022.
[12] Dabra, A., & Kumar, V. (2023). Evaluating green cover and open spaces in informal settlements of Mumbai using deep learning. Neural Computing and Applications, 1-16.
[13] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. arXiv preprint arXiv:2304.02643.
[14] Wu, Q., & Osco, L. P. (2023). Samgeo: A python package for segmenting geospatial data with the segment anything model (sam). Journal of Open Source Software, 8(89), 5663.
[15] Sentinel-2 overview. (2022).
[16] El-Asmar, H. M., & Hereher, M. E. (2011). Change detection of the coastal zone east of the Nile Delta using remote sensing. Environmental Earth Sciences, 62(4), 769-777.
[17] Tian, S., Zhong, Y., Zheng, Z., Ma, A., Tan, X., & Zhang, L. (2022). Large-scale deep learning based binary and semantic change detection in ultra-high resolution remote sensing imagery: From benchmark datasets to urban application. ISPRS Journal of Photogrammetry and Remote Sensing, 193, 164-186.
[18] Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2019). Multitask learning for large-scale semantic change detection. Computer Vision and Image Understanding, 187, 102783.
[19] Toker, A., Kondmann, L., Weber, M., Eisenberger, M., Camero, A., Hu, J., Hoderlein, A. P., Senaras, C., Davis, T., Marchisio, G., Zhu, X. X., & Leal-Taixe, L. (2022). DynamicEarthNet: Daily multispectral satellite dataset for semantic change segmentation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA.
[20] Rahnemoonfar, M., Chowdhury, T., Sarkar, A., Varshney, D., Yari, M., & Murphy, R. R. (2021). FloodNet: A high-resolution aerial imagery dataset for post-flood scene understanding. IEEE Access, 9, 89644-89654.

[21] Wang, J., Zheng, Z., Ma, A., Lu, X., & Zhong, Y. (2021). LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation. October 2021.
[22] Karra, K., Kontgis, C., Statman-Weil, Z., Mazzariello, J. C., Mathis, M., & Brumby, S. P. (2021). Global land use/land cover with Sentinel 2 and deep learning. 2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2021.
[23] Schmitt, M., Hughes, L. H., Qiu, C., & Zhu, X. X. (2019). SENI2MS - A curated dataset of georeferenced multispectral Sentinel-1/2 imagery for deep learning and data fusion. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7, 153-160.
[24] Holtz-Eakin, D., Newey, W., & Rosen, H. S. (1988). Estimating vector autoregressions with panel data. Econometrica: Journal of the Econometric Society, 1371-1395.

Early Detection of Rare Diseases using Natural Language Processing

Eglė Kondrataitė, Gražina Korvel

Vilnius University, Institute of Data Science and Digital Technologies, Akademijos str. 4 Vilnius
egle.kondrataite@mif.stud.vu.lt, grazina.korvel@mif.vu.lt

Early and accurate detection of rare diseases is an important aspect of reducing disease progression and improving the quality of life of the affected people. It is estimated that there are about 36 million people in the EU who suffer from more than 5000 rare diseases [4]. The majority of these conditions are of genetic origin and usually appear in childhood. They can often lead to disability, chronic illness, or even premature death [5]. It is difficult to accurately identify a rare disease because the symptoms can be similar to those of more widespread illnesses. Patients that are affected by such conditions constantly face delayed diagnosis, which can lead to psychological and economic challenges for them and their families [6]. Late diagnosis is associated with reduced quality of life and increased mortality rates. Therefore, early diagnosis help doctors to closely monitor the progression of the disease and avoid rapid negative changes in the patient's health.

Early detection of health risks is one of the key foundations of modern healthcare. It allows doctors to carry out necessary tests and prescribe early treatment to control the disease. However, the task of accurately and quickly diagnosing rare diseases is very challenging for general practitioners who may lack knowledge of these conditions [2]. In addition, rare diseases are under-represented in the International Classification of Diseases, version 10 (ICD-10), which is widely used for disease identification [1]. Therefore, there is a great need for new technologies to identify rare diseases.

Natural language processing (NLP) methods are rapidly gaining popularity in the medical field as electronic health records (EHRs) are increasingly implemented. These records are a rich source of data consisting of structured and unstructured information. The structured data includes the patient's medical history, diagnoses, medications, medical and surgical procedures, and allergies. The unstructured data consists of physicians' free-text notes that can include important observations about patient's

DOI: https://doi.org/10.15388/LMITT.2024.9

health. According to [3] NLP methods can be very useful for processing unstructured data in patient health records. Specifically, they have been used to uncover meaningful insights about Dravet syndrome from narrative medical reports in electronic health records [3]. In addition, NLP methods have also been used to analyze free-text clinical notes to detect depression in patients diagnosed with breast and colorectal cancer [2]. However, the application of such methods faces certain challenges. These can include data quality, medical terms in different languages, or the complexity of medical terminology in general. Some conditions may have different synonyms and abbreviations to describe them. For example, "obsessive-compulsive disorder", "anancastic neurosis", and "OCD" are the same disease. In addition, symptoms can present similar challenges where patients' complaints can be described by medical terms and also by short phrases [2]. However, the main difficulty is data annotation. In order to apply machine learning (ML) algorithms, the data must be labelled. However, annotating rare diseases in clinical notes requires expertise in specific fields, hence significant cost and time from clinical experts [1]. Another scientific challenge is that most rare diseases have a limited number of cases and may present with unusual symptoms. Therefore, the development of NLP models that can handle unique linguistic features and terminology of rare diseases is necessary.

Keywords: Early Detection, Rare Diseases, Natural Language Processing, Electronic Health Records

References

[1] Hang Dong, Victor Suarez-Paniagua, Huayu Zhang, Minhong Wang, Emma Whitfield, and Honghan Wu. Rare disease identification from clinical notes with ontologies and weak supervision. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pages 2294–2298. IEEE, 2021.

[2] Angela Leis, David Casadevall, Joan Albanell, Margarita Posso, Francesc Macia, Xavier Castells, Juan Manuel Ramirez-Anguita, Jordi Martinez Roldan, Laura I. Furlong, Ferran Sanz, et al. Exploring the association of cancer and depression in electronic health records: Combining encoded diagnosis and mining free-text clinical notes. JMIR Cancer, 8(3): e39003, 2022.

[3] Tommaso Lo Barco, Nicolas Garcelon, Antoine Neuraz, and Rima Nabbout. Natural history of rare diseases using natural language processing of narrative unstructured electronic health records: the example of dravet syndrome. Epilepsia, 65(2):350–361, 2024.

[4] Julio Lopez-Bastida, Juan Oliva-Moreno, Renata Linertova, and Pedro Serrano-Aguilar. Social/economic costs and health-related quality of life in patients with rare diseases in europe. The European Journal of Health Economics, 17(Suppl 1):1–5, 2016.

[5] Shruti Marwaha, Joshua W. Knowles, and Euan A. Ashley. A guide for the diagnosis of rare and undiagnosed disease: beyond the exome. Genome Medicine, 14(1):23, 2022.
[6] Yvonne Zurynski, Marie Deverell, Troy Dalkeith, Sandra Johnson, John Christodoulou, Helen Leonard, Elizabeth J. Elliott, and APSU Rare Diseases Impacts on Families Study group. Australian children living with rare diseases: experiences of diagnosis and perceived consequences of diagnostic delays. Orphanet Journal of Rare Diseases, 12:1–9, 2017.

Mašininio mokymosi pritaikymas reklamų aptikimui YouTube įrašuose

Karolis Kvedaravičius, Olga Kurasova

Vilniaus universitetas, Duomenų mokslo ir
skaitmeninių technologijų institutas, Akademijos g. 4, Vilnius
karolis.kvedaravicius@mif.stud.vu.lt

Santrauka. Šiame straipsnyje aprašyta kaip tyrimų metu buvo bandoma pritaikyti mašininį mokymąsi reklamų aptikimui *YouTube* vaizdo įrašuose naudojant transkribuotą tekstą. Reklamų aptikimas buvo laikomas teksto klasifikavimo užduotimi ir todėl buvo naudojamas BERT šeimos mašininio mokymosi modelis, kuris pasiekia aukštus rezultatus sprendžiant teksto analizės uždavinius. Tačiau šiam modeliui dėl įvairių priežasčių buvo sunku pasiekti aukštą tikslumo lygį. Bet naudojant antrą straipsnyje pasiūlytą klasifikavimo žingsnį, kuris atsižvelgia į BERT modelio klasifikavimą tam tikram laiko tarpe, rezultatai buvo pagerinti.

Raktiniai žodžiai: Mašininis mokymasis, reklamos, transkribuotas tekstas.

1 Įvadas

Šiais laikais *YouTube* vaizdo įrašų turinyje yra dažnai įterpiamos reklamos, kuriose įrašo kūrėjas perduoda informaciją iš rėmėjų (angl. *sponsor segments*). Šios reklamos ne visada yra aktualios žiūrovams. Jau dabar yra sistemų (viena iš jų yra *SponsorBlock*), kurios, naudodamos vartotojų įkeltus duomenis, žymi ir praleidžia reklamas *YouTube* vaizdo įrašuose. Tačiau ši sistema turi trūkumų, reikia palaukti kol kiti vartotojai sužymės reklamas ir jei įrašas neturi žiūrovų, naudojančių *SponsorBlock,* reklamos niekad nebus aptiktos.

Todėl atrodė vertinga bandyti sukurti automatizuotą sistemą, kuri atliktų reklamų aptikimo procesą automatiškai, pasitelkiant mašininį mokymąsi. Kadangi galima pasiekti *YouTube* įrašų transkribuotą tekstą per *YoutubeTransciptAPI*, šį tekstą galima klasifikuoti naudojant mašininio mokymosi modelius. Šio tyrimo tikslas apmokyti mašininio mokymo modelį reklamų aptikimo uždaviniui naudojant transkribuoto tekstą ir pateikti modelio rezultatus vykdant reklamų aptikimo uždavinį.

DOI: https://doi.org/10.15388/LMITT.2024.10

2 Literatūros apžvalga

Kadangi transkribuotame tekste reklamų paieška yra teksto klasifikavimo uždavinys, buvo nagrinėjama, kokie modernūs modeliai yra tinkami atliekant teksto klasifikavimą. 2017 metais buvo pasiūlytas naujas mašininio mokymo modelis – transformeris [1]. Jis išsiskyrė iš kitų modelių, nes savo architektūroje naudoja dėmesio (angl. attention) mechanizmą, vietoje konvoliucinių sluoksnių.

Transformerių architektūra buvo pritaikyta BERT (angl. *Bidirectional Encoder Representations from transformers*) modeliams. BERT buvo sukurtas 2018 Google mokslininkų [2]. BERT yra iš anksto apmokytas (angl. *Pre-trained*) modelis, kuris apmokytas MLM (angl. *masked language modeling*) ir NSP (angl. *Next Sentence Prediction*) uždaviniams spręsti. BERT modelis gali būti pritaikytas įvairiems natūralios kalbos apdorojimo uždaviniams. Pavyzdžiui, GLUE (angl. *General Language Understanding Evaluation*) pasiekia 80 % įvertinimą.

Taip pat BERT modelis gali būti sėkmingai pritaikytas ne tik tekstų anglų kalba analizei. Yra sukurtas lietuvių ir latvių kalbomis apmokytas BERT modelis, kuris pasiekia aukštesnius rezultatus negu daugiakalbis BERT modelis, taikomas lietuvių ir latvių kalbų uždaviniams [3].

3 Duomenų surinkimas

Kadangi nėra tinkamos viešos duomenų aibės apmokyti BERT modelį reklamų aptikimui *YouTube* įrašuose užduočiai, šio tyrimo eigoje buvo sudaryta nauja duomenų aibė. Duomenų aibė buvo surinkta naudojant *SponsorBlock* atviro kodo duomenų bazę ir *YoutubeTranscriptAPI*. Iš *SponsorBlock* paimamas įrašo URL (kad vėliau būtų galima paimti iš *YoutubeTranscriptAPI* transkribuotą tekstą) *startTime* – reklamos pradžios laiką, *endTime* – reklamos pabaigos laiką, *votes* – vartotojų įvertinimą, *type* – turinio tipą. Kadangi duomenų kiekis didelis ir jų visų negalima patikrinti, pasirenkami tik tie įrašai, kurių tipas *sponsor* ir turi daugiau negu 100 teigiamų įvertinimų.

Antra naudojant *YoutubeTranscriptAPI* ir URL paimamas įrašo transkribuotas tekstas. Grąžinamas tekstas yra suskirstytas į atkarpas (vidutiniškai 40 simbolių ilgio). Taip pat grąžinama, kurioje įrašo sekundėje atkarpa baigiasi ir kiek sekundžių atkarpa tęsiasi.

Tada *SponsorBlock* duomenų bazės atrinktos reklamos ir iš *YoutubeTranscriptAPI* surinktas tekstas naudojami sudaryti mokymosi duomenis su

klasėmis. Kiekvienos transkribuoto teksto atkarpos pradžios ir pabaigos laikai lyginami su iš *SponsorBlock* duomenų bazės surinktais reklamų pradžios ir pabaigos laikais. Jei teksto atkarpos pradžios arba pabaigos laikas įkrenta tarp reklamos pabaigos ir pradžios laiko, transkribuotas tekstas priskiriamas klasei 1 (klasė 1 žymi atkarpas, kurios yra reklamos), kitu atveju priskiriama klasei 0 (klasė 0 žymi atkarpas, kurios nėra reklamos).

Tokiu metodu iš viso buvo surinkta 741815 transkribuoto teksto eilučių. 700697 teksto eilučių priklausė klasei 0, 41118 eilučių priklausė klasei 1. Klasei 1 priklausė tik 5 % visų surinktų eilučių. Surinktų duomenų pavyzdys yra pateiktas 1 lentelėje.

Šis būdas rinkti duomenis iškelia kelias problemas. Kadangi *YouTube* įrašų kūrėjai po įkėlimo turi kelis būdus modifikuoti jau įkeltą įrašą. Vienas iš jų yra iškirpimas tam tikro fragmento įrašo. Dėl to gali būti, kad *SponsorBlock* duomenų bazėje yra pažymėta reklama, kuri naujoje įrašo versijoje neegzistuoja. Taip pat surinkti duomenys yra priklausomi, nuo kokius vaizdo įrašus žiūri *SponsorBlock* vartotojai ir kaip tiksliai yra pažymėtos reklamos šių vartotojų.

1 lentelė. Surinktų duomenų pavyzdys.

Nr.	Tekstas	Klasė
1	luck cause this isn't even the hardest part yet	0
2	just before this video gets going I want	1
3	to give a special thanks and mention to	1
4	nitrous networks our server provider for	1

4 Modelis reklamoms klasifikuoti

Šiame skyriuje aprašomas BERT apmokymas su sudaryta duomenų aibe ir apmokyto BERT modelio rezultatai sprendžiant reklamų aptikimo užduotį. Taip pat aprašyta papildomo klasifikavimo žingsnio veikimas ir rezultatai. Papildomas klasifikavimo žingsnis buvo naudojamas norint pasiekti aukštesnius klasifikavimo rezultatus.

4.1 BERT modelio apmokymas

Reklamų aptikimo iš transkribuoto teksto užduočiai buvo apmokytas *BertForSequenceClassification* modelis, kuris yra BERT modelis, specialiai pritai-

kytas klasifikavimo uždaviniams su papildomais išmetimo ir klasifikavimo sluoksniais. Kadangi BERT modelio apmokymas su pilna sudaryta duomenų aibe užtruktų ilgai, tyrimo metu modelis buvo apmokytas su mažesniu duomenų kiekiu. Apmokymas atliktas su dviem duomenų kiekiais, 100 tūkstančių ir 200 tūkstančių teksto eilučių norint ištirti duomenų kiekio įtaką BERT modelio rezultatams. Taip pat ištestuoti dveji skirtingi klasių balansai, originalus: 95 % - klasė 0, 5 % - klasė 1 ir pakeistas: : 85 % - klasė 0, 15 % - klasė 1, ir to poveikis vertinimo rezultatams. Vertinimo duomenų aibę sudarė 30 tūkstančių teksto eilučių.

Visiems modelio mokymams buvo naudotas tie patys hiperparametrai, kurie pateikti 2 lentelėje.

2 lentelė. Mokymui naudoti hiperparametrai

Partijos dydis	64
Epochų skaičius	4
Mokymosi greitis	$2{,}5e^{-5}$
Maksimalus įvesties ilgis	128

Visų atliktų apmokymų rezultatai pateikti 3–5 lentelėse. Visose lentelėse pateikti geriausių epochų rezultatai. Geriausias bendras tikslumas (angl. *accuracy*) gautas, kai modelis apmokomas su 100 tūkstančių eilučių su originaliu klasių balansu. Tačiau, metrikų reikšmės klasei 1 (reklamos) yra žemos palyginus su klasės 0. Pakeičiant klasių balansą į 85 % - klasė 0, 15 % - klasė 1, metrikos klasei 1 pagerėja, tačiau bendras tikslumas nukrenta. Padvigubinus eilučių kiekį apmokymui rezultatai pagerėja tik labai nežymiai.

BERT modeliui gali būti sunku teisingai klasifikuoti klasę 1, dėl šios klasės retumo palyginus su klase 0. Kita problema galėtų būti transkribuoto teksto eilučių ilgis. Eilučių ilgis vidutiniškai yra 40 simbolių ir galimai modelis gauna nepakankamai informacijos, kad galėtų teisingai klasifikuoti.

3 lentelė. Mokymosi rezultatas su 95000 teksto eilučių priklausant klasei 0, 5000 priklausant klasei 1.

Bendras tikslumas	**Tikslumas**	**Atpažinimas**	**F1-balas**	**Klasė**
0,95	0,96	0,99	0,97	0
	0,57	0,29	0,39	1

4 lentelė. Mokymosi rezultatas su 85000 teksto eilučių priklausant klasei 0, 15000 priklausant klasei 1.

Bendras tikslumas	Tikslumas	Atpažinimas	F1-balas	Klasė
0,92	0,97	0,95	0,96	0
	0,37	0,53	0,44	1

5 lentelė. Mokymosi rezultatas su 170000 teksto eilučių priklausant klasei 0, 30000 priklausant klasei 1.

Bendras tikslumas	Tikslumas	Atpažinimas	F1-balas	Klasė
0,93	0,97	0,96	0,96	0
	0,41	0,45	0,44	1

4.2 Papildomas klasifikavimo žingsnis

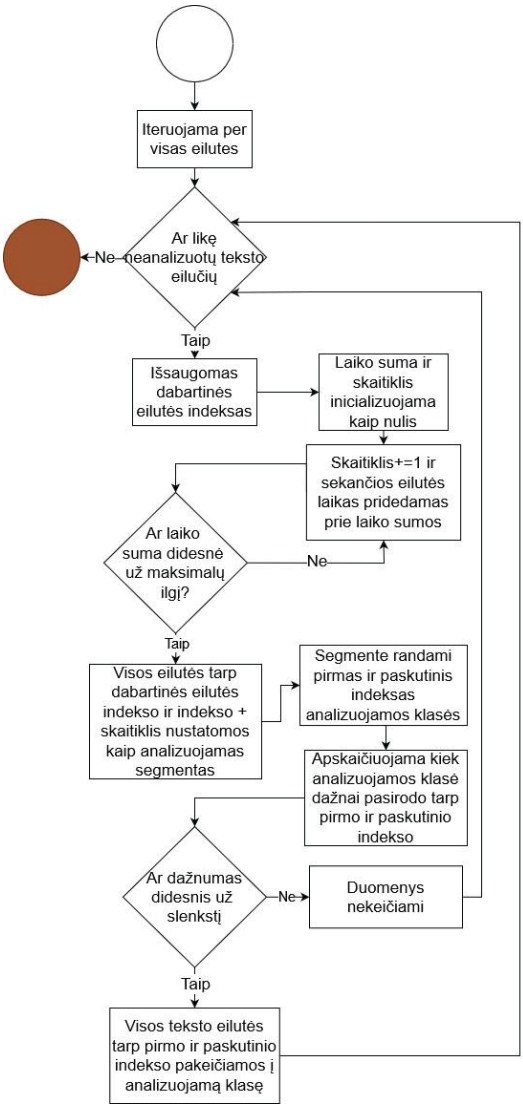
Peržiūrėjus kaip modelis klasifikuoja *YouTube* įrašų teksto eilutes pastebėta, kad dažnai pasikartoja dviejų tipų klaidos. *YouTube* įrašuose reklamos dažniausiai susideda iš kelių, viena po kitos einančių eilučių. Tačiau BERT modeliui sunku klasifikuoti visas reklamos eilutes. Reklamose atsiranda „tarpai“, kur neteisingai klasifikuojamos reklamos eilutės kaip ne reklamos. Gali būti, kad „tarpai“ atsiranda, nes teksto eilutės yra trumpos (maždaug 40 simbolių ilgio) ir jose kartais nėra pakankamai informacijos, kad modelis teisingai klasifikuotų.

Taip pat, kartais BERT modelis neteisingai klasifikuoja pavienes eilutes kaip reklamas, nors jos nėra reklamos ir yra apsuptos ne reklamos eilučių. Šio tipo klaidos vadinamos triukšmu. 1 pav. pavaizduota kaip šios klaidos atrodytų pažymėtos *YouTube* vaizdo įrašo laiko juostoje.

1 pav. *YouTube* vaizdo įrašo teksto klasifikavimo klaidų pavyzdys

Norint sumažinti šių problemų įtaką rezultatams buvo kuriamas papildomas klasifikavimo algoritmas, kuris darė dvi prielaidas. Pirma, kad *YouTube* įrašuose reklamos yra vientisos, reklamos nebus pertrauktos trumpų sakinių, kurie nėra reklamos dalis. Antra, reklamos yra tam tikro minimalaus

ilgio, jei eilutė, modelio pažymėta kaip reklama, bus apsupta eilučių kurios nėra reklama tai bus laikoma klaida.



2 pav. Papildomas klasifikavimo algoritmas

Pagal idėją algoritmas panašus į vaizdų analizėje dažnai naudojamas morfologines operacijas: eroziją (angl. *erosion*) ir plėtimą (angl. *dilation*). Realizuojant algoritmą naudojama iš *YouTubeTranscriptAPI* gaunama informacija. *YouTubeTranscriptAPI* grąžina ne tik kiekvienos įrašo eilutės tekstą, bet ir eilutės trukmę sekundėmis. Kiekvienos eilutės trukmė naudojama nustatyti, kiek aplinkinių eilučių algoritmas analizuoja, norint nuspręsti ar apmokytas BERT modelis klaidingai klasifikavo eilutę. Algoritmas nusprendžia, kad BERT modelis klaidingai klasifikavo teksto eilutę, priklausomai nuo kaip dažnai pasikartoja tam tikra klasė analizuojamame segmente.

Algoritmo veikimas pavaizduotas 2 pav. Algoritmas turi keturis įvesties duomenis:

- Sąrašas BERT modelio kiekvienos eilutės klasifikavimo.
- Maksimalus analizuojamo segmento ilgis, kuris nustato kiek eilučių analizuojama vienu metu.
- Slenkstis, kuris nustato ar tam tikra klasė pasikartoja analizuojamame segmente pakankamai retai, kad tai būtų laikoma klaida.
- Analizuojama klasė. Jei analizuojama klasė 1 atliekamas „tarpų“ šalinamas. Jei analizuojama klasė 0 atliekamas „triukšmo“ šalinamas.

6 lentelė je pateikti vertinimo rezultatai pridėjus papildomą klasifikavimo žingsnį (BERT modelis apmokytas su 200 tūkstančių eilučių, 85 % - klasė 0, 15 % - klasė 1). Bendras tikslumas pagerėja nuo 0,93 iki 0,95 ir žymiai pagerėja F1-balas klasei 1, nuo 0,44 iki 0,55.

6 lentelė. Rezultatai naudojant papildomą klasifikavimo žingsnį

Bendras tikslumas	Tikslumas	Atpažinimas	F1-balas	Klasė
0,95	0,97	0,98	0,98	0
	0,61	0,5	0,55	1

5 Išvados

Tyrimo metu buvo sudaryta nauja duomenų aibė, kurioje *YouTube* vaizdo įrašų transkribuotas tekstas suskirstytas į reklamas ir ne reklamas. Duomenų aibė sudaryta naudojant viešai pasiekiamus duomenis iš *SponsorBlock* ir *YouTubeTranscriptAPI.* Iš viso surinkta apie 740 tūkstančių eilučių. Su šia duomenų aibe buvo apmokytas BERT modelis reklamų aptikimo uždaviniui. BERT modelis pasiekia gana aukštą bendrą tikslumą klasifikuojant *YouTube*

vaizdo įrašų transkribuotą tekstą į reklamas ir ne reklamas. Tačiau rezultatai reklamos klasei yra gana žemi, F1-balas yra tik 0,45, o ne reklamos klasė F1-balas yra 0,97. Pridėjus antrą klasifikavimo žingsnį, kuris atsižvelgia į BERT modelio klasifikavimą tam tikram laiko tarpe, rezultatai pagerėja, ypač reklamos klasei. Reklamos klasės F1-balas pagerėja nuo 0,45 iki 0,55.

Literatūra

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, „Attention Is All You Need," p. 15, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Minneapolis, 2019.

[3] Matej Ulčar, Marko Robnik-Šikonja, „Training dataset and dictionary sizes matter in BERT models: the case of Baltic languages," Lublianos Universitas, Lubliana, 2021.

The Influence of YOLOv5 Hyperparameters for Construction Details Detection

Tautvydas Kvietkauskas

Vilnius Gediminas Technical University, Department of Information Technology, Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
tautvydas.kvietkauskas@stud.vilniustech.lt

Abstract. Computer vision has become a fundamental area of interest in recent decades. Each area has unique data which object detection methods can analyse. However, it is important to find the most suitable parameters for the model that detects different object groups. In this research has been investigated the influence of pre-trained YOLOv5 (nano (n), small (s), medium (m), large (l), extra-large (x)) models, hyperparameters (learning rate, momentum, and weight decay) and different image augmentation (hsv_h, degrees, translate, flipud, mosaic, mixup, shear, perspective) efficiency for similar construction details detection. A newly collected dataset with twenty-two labelled categories of construction details was prepared. A total of 270 models were trained and evaluated. Every model was evaluated with 3,300 test images which backgrounds were mixed, neutral, and white backgrounds. The most accurate model was YOLOv5l with learning rate – 0.001, momentum – 0.950 and weight decay – 0.0001. This model achieved – 0.5015 (50.15%) accuracy.

Keywords: YOLOv5, object detection, hyperparameters, constructions details.

1 Introduction

Computer vision is a rapidly developing field of artificial intelligence designed to enable machines to interpret and understand visual information from the surrounding environment. By simulating the human visual system, computer vision systems can extract meaningful insights from images and videos, revolutionising various industries. From house number recognition [17] to medical pills [1] and drones [2], computer vision algorithms play an important role in analysing and interpreting visual data.

To get the highest object detection results, it is necessary to have the proper data set. The most difficult problem in object recognition is similar-looking objects. Similarity can be seen in shape, colour, and size. Object detection, which is used to detect what fruit [3] is bought at self-service

DOI: https://doi.org/10.15388/LMITT.2024.11

checkouts, can easily make mistakes regarding the type of apples because of similarities in appearance. Image shooting angles, shadows, lighting, distance, and other additional factors make different types of objects look the same. The same problem exists when trying to detect construction details.

This study looked at several types of pre-trained YOLOv5 models using newly gathered construction detail datasets [12]. The training dataset includes 440 photos (22 categories, each with 20 images). For testing, 3300 photos (22 construction details on mixed, neutral, and white backgrounds; 50 photographs for each group). The experimental inquiry was divided into two stages: main and additional. In the main

stage, 135 experiments were carried out using the YOLOv5 models nano, small, medium, large, and extra-large. The optimal learning rate, weight decay, and momentum were observed. According to the main stage learning curves and accuracy, an additional 135 models were built and evaluated for potential accuracy improvements. The originality of this work is a thorough investigation of 270 experiments in which various YOLOv5 hyperparameters were analysed. The findings may be useful in other applications requiring the detection of similar feature items. Furthermore, experimental results can help build the construction recommendation model. It can be practically applied in a smartphone app that suggests various constructions based on observed details in real time.

2 Objects Detection Methods Review

Popular object detection models are SSD [4], Faster R-CNN [5] and YOLO group [6-10]. SSD as a single-shot detector, efficiently predicts bounding boxes and class probabilities simultaneously, striking a balance between speed and accuracy suitable for real-time applications. In contrast, Faster R-CNN adopts a two-stage architecture, leveraging a Region Proposal Network (RPN) to generate region proposals before refining and classifying them. While offering higher accuracy, this approach sacrifices speed and is more suitable for tasks requiring precision. YOLO predicts bounding boxes and class probabilities for each grid cell, making it incredibly fast and ideal for real-time applications, particularly for small objects.

Three different object detection methods have been examined using medication pills. Correct identification is essential for safe medicine administration. In a real-time pill recognition investigation, Faster R-CNN,

SSD, and YOLOv3 recognition algorithms were employed to assess recognition accuracy and speed. The tablets were randomly arranged, and 5,131 photos were captured. The dataset contains 70 capsules and 191 non-capsules. The training parameters for each algorithm have been adjusted to 64 batches, 16 sub-divisions, 0.001 learning rate, 0.9 momentum, and 0.0001 weight decay. Based on these findings, researchers determined that YOLOv3 is faster than SSD and Faster-R-CNN. According to the mAP indication, Faster R-CNN appears to be the highest (82.89%), however, its detection rate is just 17 frames per second. The SSD-based model achieved an average of 32 frames per second and 82.71% mAP. Compared to recent models, the YOLOv3 achieves only 80.69% mAP, but it can significantly improve detection rates and attain real-time performance at 51 frames per second. As a result, it was determined that the YOLO group model would be appropriate for real-time pill detection because it can recognize pills quickly and with reasonable accuracy [1].

To efficiently use recognition to identify road traffic items, the optimum object identification method must be discovered. Many object identification algorithms have recently been released, although there is little material comparing algorithms, such as YOLOv5, which is focused on road traffic objects. The article investigates SSD MobileNetv2, YOLOv3, YOLOv4, and YOLOv5 for real-time street-level item recognition. The dataset comprised 3,169 pictures with 24,102 annotations. Five classes were identified: automobiles (16446 comments), traffic lights (4790), crossings (1756), trucks (761), and motorcyclists (349). The dataset was separated into three parts: training (2010), validation (586), and testing (573). Each image was rescaled with HSV scaling (-25 to 25), noise augmentation (up to 5% of pixels), and cut-out (3 cells at 10% each) was also used. During the training phase of YOLO group algorithms, the SGD optimizer was set together with 100 epochs. Meanwhile, 32000 training steps were scheduled for the SSD MobileNetv2 FPN-lite. YOLOv4 had comparatively lower F1 scores, accuracy, and mAP than YOLOv5l and YOLOv3. The data suggest that YOLOv5l is the most accurate (Precision – 0.780) algorithm for this experiment. However, when compared to the other YOLO models, the mAP rates were not significantly different (SSD – 0.315, YOLOv3 – 0.313, YOLOv4 – 0.304, YOLOv5l – 0.313, YOLOv5s – 0.260). Also, YOLOv4 was the slowest of the models. Meanwhile, YOLOv5 performs better than previous YOLO versions in terms of mAP@.5 and inference time. SSD MobileNetv2 FPN-lite had the lowest mAP@.5

performance of any of the object identification algorithms tested in this trial, with a score of 0.315. However, it is the fastest algorithm in the trial, taking 6.3 milliseconds. The second quickest object detection technique is YOLOv5s – 8.50 milliseconds, an F1-Score of 0.579, and a mAP@.5 of 0.530, which is just 11% worse, and mAP@.5:95 is 17% poorer than YOLOv5l, the most accurate model in this experiment. In conclusion, YOLOv5l is the most accurate algorithm [8].

According to other research about SSD, YOLO, and Faster R-CNN, it was decided to choose YOLOv5 due to the training time and accuracy ratio, the lowest detection loss, prediction time and the best stability in the YOLO group.

3 Analysis of New Dataset

The first dataset of four [12] has been used only for the training phase. Each of the different twenty-two classes of details was photographed only on a white background (Figure 1, first image). Every detail was rotated twenty times and photos from the new angle were taken. The training data set contains 440 images because pre-trained [13, 14] YOLOv5 models were used. For the testing dataset, each construction detail was photographed from fifty different angles on white (W), neutral (N) and mixed (M) backgrounds (Figure 1, second image). These three different backgrounds simulated the possible real-world environment. In general, 1100 images for each background have been prepared, in total 3300 different images.

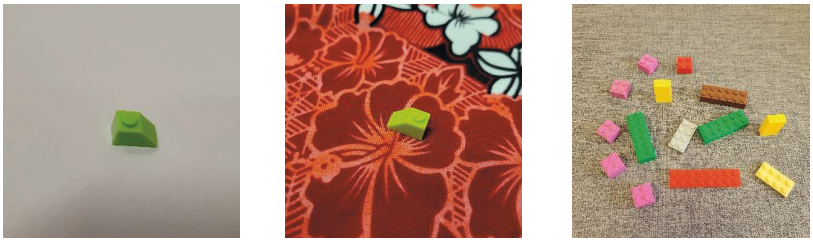


Figure 1. The samples of the datasets.

4 Experiment Methodology

Investigation of various hyperparameters efficiency for the accuracy of YOLOv5 has been done. In the main research, 135 models were trained and evaluated to find the highest accuracy in construction details detection.

additional research in which other 135 models were trained and evaluated according to main research training specifications, statistics and learning curves. The research workflow is in Figure 2.

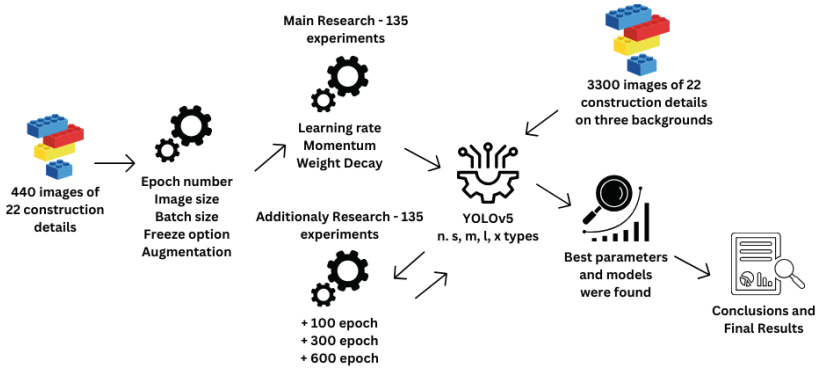


Figure 2. The methodology of the experimental investigation.

The main stage of experiments was focused on hyperparameters, while the additional stage was for additional epochs. For every experiment, pre-trained YOLOv5 [13] was used. The models which were used are already pre-trained with the COCO2017 dataset [14]. The dataset contains 164,000 labelled images of 80 different objects. The models have been trained using 118,000 images, for validation of 5,000 images and testing 41,000 images. During the experiment, every model was trained using Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz (20 Threads, 10 Cores). Hardware has been used Linux operating system with 32-GB DDR4 RAM and GPU - Tesla P100 PCIe 12GB.

The mean Average Precision (mAP) with a predefined IoU (Intersection Over Union) threshold usually evaluates object detection results during the training stage. Across experiments, models with low accuracy had low mean Average Precision (mAP), and vice versa. After examining the learning curves, it was decided to select models based on correct detection accuracy. This is because models with similar training results showed significantly different accuracy. Differences between models ranged from 100 to 200 detected construction details after testing.

5 Results of the Main and Additional Research

Based on other research [17-19] and our pilot studies, the results have shown that augmentation has a positive impact on detection accuracy. Furthermore, experiments have shown that by freezing backbones, the accuracy increases about 1.5 times. According to the overall results, the detection accuracy is much better on a neutral background. However, all models have been trained with images in which the background was only white. In general, the pilot parameters became like this: image size – 320, batch size – 32, epoch number – 300, layers freeze option – 10, hsv_h – 0.09, hsv_s – 0.7, hsv_v – 0.4, degrees – 0.125, translate – 0, scale – 0.5, shear – 0.9, perspective – 0, flipud – 0.5, fliplr – 0.5, mosaic – 0, mixup – 0, copy_paste – 0.

The analysis of related works showed that most researchers focus on learning rate, momentum, and weight loss [16, 17]. In the main research, the nano (n), small (s), medium (m), large (l), and extra-large (x) versions of the YOLOv5 have been trained using the parameters of the pilots research. A total of 270 models were trained and evaluated. The parameters used in the main research: learning rate – 0.01, 0.001, 0.0001; momentum – 0.9, 0.937, 0.95; weight decay – 0.0001, 0.0005, 0.0007. The other values of the parameters were default. After the main research (135 trained models), additional pieces of training were done because according to results, and learning charts, some models are underfitting. Therefore, models that trained with a 0.01 learning rate were trained additionally with 100, 0.001 - 300 and 0.0001 - 600 epochs. The results have shown that the highest accuracy of the main research is equal to 0.5012 (50.12%). It was achieved with the YOLOv5l model with a learning rate equal to 0.001, a momentum is 0.95, and a weight decay is 0.0007. In some cases, the correct detection ratio was equal to 0. It happens because of the too short training time. Therefore, additional trainings were made. However, for models whose accuracy was 0, it became higher but overall did not make much of an impact. The highest accuracy additional trained model was also YOLOv5l, learning rate - 0.001 and momentum - 0.950; however, the weight decay is 0.0001. This model achieved slightly better results – 0.5015 (50.15%). On the other hand, the most accurate models of YOLOv5 nano (n), medium (m), and extra-large (x) achieved slightly lower results than models of the main experiment (Figure 3).

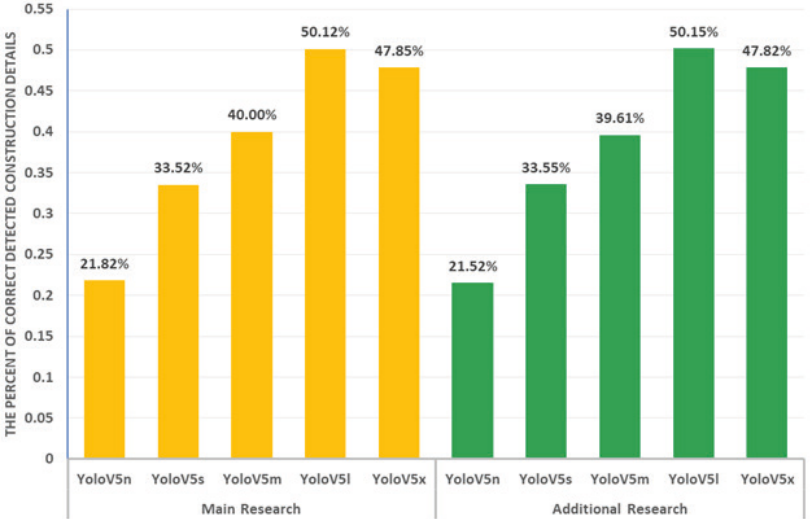


Figure 3. The percentage of correct detection ratio of each nano, small, medium, large, and extra-large model.

As Figure 3 illustrates, according to the main research, YOLOv5n, which has the lowest number of CNN, shows the lowest precision in the detection ratio – 21.82%, while YOLOv5x with the highest amount of CNN achieves 47.85%. The different results for YOLOv5s (33.52%) and YOLOv5m (40%) is 6.48%. Similar ratios were achieved after additional experiments. YOLOv5 nano (n), medium (m) and extra-large (x) have slighter lower accuracies, while small (s) and large (l) versions achieved slighter higher accuracies. In both experiments, YOLOv5l showed the highest results. The main research YOLOv5l – 1654 (mix – 497, neutral – 562, white – 595), while the additional research YOLOv5l – 1655 (mix – 496, neutral – 561, white – 598).

6 Conclusions

This study examined the impact of the training parameters and hyperparameters on the identification of construction details. When analysing similar feature data, the task complexity led to the selection of construction details. Recognition is dependent on the shot's angle, which is determined by the camera's point of view. Throughout the study, the five

pre-trained YOLOv5 models were examined. In total, 270 models have been trained and evaluated. Three different complexity backgrounds containing a total of 3300 photos were used to assess the efficiency models. Learning rate, momentum, and weight decay were examined. Every parameter was used in various combinations. According to the findings of the experimental investigation, coloured images, an image size of 320, a batch size of 32, epoch number of 300, an option of layer freeze of 10, data enhancement is used, learning rate of 0.001, momentum of 0.95, and a weight decay of 0.0007 are the optimal parameters for the detection of construction details. Another optimal parameter with almost similar accuracy can be the same as it was mentioned but with a learning rate – 0.0001 and 900 epochs. Regardless of the background chosen, the proportion of proper detection in this case is ~ 50%. The results of the experimental investigation indicate that the use of a mixed background yields the least detection results. The primary cause is that some details become lost in the background, making it impossible for the models to identify any details at all.

References

[1] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," BMC Medical Informatics and Decision Making, vol. 21, no. 1, Nov. 2021, doi: 10.1186/s12911-021-01691-8.

[2] S. M. Alkentar, B. Alsahwa, A. Assalem, and D. Karakolla, "Practical comparison of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection," Mağallaẗ Al-handasaẗ, vol. 27, no. 8, pp. 19–31, Aug. 2021, doi: 10.31026/j.eng.2021.08.02.

[3] Hameed, K.; Chai, D.; Rassau, A. A sample weight and adaboost cnn-based coarse to fine classification of fruit and vegetables at a supermarket self-checkout. Applied Sciences, 2020, 10(23), 8667.

[4] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Lecture Notes in Computer Science, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.

[6] C. Li et al., "YOLOV6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv.org, Sep. 07, 2022. https://arxiv.org/abs/2209.02976

[7] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv.org, Jul. 06, 2022. https://arxiv.org/abs/2207.02696

[8] U. Nepal and H. Eslamiat, "Comparing YOLOV3, YOLOV4 and YOLOV5 for autonomous landing spot detection in faulty UAVs," Sensors, vol. 22, no. 2, p. 464, Jan. 2022, doi: 10.3390/s22020464.

[9] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," arXiv.org, May 17, 2023. https://arxiv.org/abs/2305.09972
[10] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOV9: Learning what you want to learn using programmable gradient information," arXiv.org, Feb. 21, 2024. https://arxiv.org/abs/2402.13616
[11] J. Kim, J.-Y. Sung, and S.-H. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), Nov. 2020, doi: 10.1109/icce-asia49877.2020.9277040.
[12] Data set of Construction details. https://app.box.com/s/j420ld0wo89hvh6np1rc3z-9t1e65yg2k.
[13] Jocher, G. YOLOv5 by Ultralytics (Version 7.0), Computer software, 2020, https://doi.org/10.5281/zenodo.3908559.
[14] T.-Y. Lin et al., "Microsoft COCO: Common Objects in context," arXiv.org, May 01, 2014. https://arxiv.org/abs/1405.0312.
[15] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly Efficient Salient Object Detection with 100K Parameters," arXiv.org, Mar. 12, 2020. https://arxiv.org/abs/2003.05643
[16] K. Nakamura and B.-W. Hong, "Adaptive weight decay for deep neural networks," IEEE Access, vol. 7, pp. 118857–118865, Jan. 2019, doi: 10.1109/access.2019.2937139.
[17] M. Taşyürek and C. Öztürk, "A fine-tuned YOLOv5 deep learning approach for real-time house number detection," *PeerJ*, vol. 9, p. e1453, Jul. 2023, doi: 10.7717/peerj-cs.1453.
[18] Y. Y. Liau and K. Ryu, "Status recognition using Pre-Trained YOLOV5 for Sustainable Human-Robot Collaboration (HRC) system in Mold assembly," *Sustainability*, vol. 13, no. 21, p. 12044, Oct. 2021, doi: 10.3390/su132112044.
[19] T. Kvietkauskas and P. Stefanovič, "Influence of training parameters on Real-Time similar object detection using YOLOV5S," *Applied Sciences*, vol. 13, no. 6, p. 3761, Mar. 2023, doi: 10.3390/app13063761.

Duomenų augmentacijos naudojant generatyvinį besivaržantį tinklą saulės kolektorių segmentavimui iš nuotolinio stebėjimo vaizdų

Justinas Lekavičius

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Naugarduko g. 24, LT-03225 Vilnius
me@justinaslekavicius.com

Santrauka. Populiarėjant saulės baterijų naudojimui didėja ir duomenų poreikis planavimui bei valdymui. Deja, šie duomenys sunkiai prieinami arba neegzistuoja, o resursai, skirti segmentavimo modelių mokymui, yra apriboti ir plėtimui reikia daug išteklių. Šiame darbe panaudotas pix2pix generatyvinis besivaržantis tinklas naujų nuotraukų generavimui iš turimų duomenų, padidinant mokymo išteklių DeepLabV3 segmentavimo modeliui kiekį. Naudojant žinių perkėlimą, modelio adaptavimą bei 60% sugeneruotų nuotolinio stebėjimo nuotraukų kaip papildomus mokymo duomenis, padidintas aptiktų kolektorių kiekis, modelio tikslumas (angl. accuracy) padidintas 0.78%, taiklumas (angl. precision) – 3.41%, jautrumas (angl. sensitivity) – 2.49%, F1 metrika – 2.71%, IoU (intersect over union) metrika – 3.19%, o nuostoliai (angl. loss) sumažėjo 0.0282.

Raktiniai žodžiai: gilusis mokymasis, saulės kolektoriai, semantinis segmentavimas, duomenų augmentacijos, generatyviniai besivaržantys tinklai, nuotolinis stebėjimas, žinių perkėlimas.

1 Įvadas

Atsinaujinantys ištekliai, ypač saulės energija išsiskiria savo švarios ir prieinamos elektros gamybos galimybėmis. Saulės baterijų diegimo augimas taip pat kelia duomenų, tokių kaip tikslių kolektorių vietovių, tipų ir specifikacijų, paklausą efektyviam planavimui ir valdymui [1]. Visgi šie duomenys riboti, o jų trūkumui mažinti reikia laiko ir pastangų. Nuotolinio stebėjimo ir giliojo mokymosi derinys iškyla kaip sprendimas, naudojant palydovų vaizdus saulės kolektorių aptikimui ir analizei. Semantinis segmentavimas, naudojant konvoliucinius neuroninius tinklus, tokius kaip FCN [2] ir U-Net [3], leidžia tiksliai identifikuoti saulės kolektorius. Be to, pažangesni mode-

DOI: https://doi.org/10.15388/LMITT.2024.12

liai, tokie kaip RU-Net [4], ir hierarchiniai metodai, naudojant EfficientNet-B5 tinklą [5], didina aptikimo tikslumą. Nepaisant pažangos, kyla iššūkis dėl sužymėtų duomenų stokos ir yra poreikis įvairesniems duomenų rinkiniams efektyviam modelio mokymui [6]. Duomenų augmentacija tampa ypač svarbi, o klasikinės augmentacijos bei generatyvinių besivaržančių tinklai [7] atlieka svarbų vaidmenį didinant duomenų kiekį. Šie tinklai, ypač pix2pix [8], suteikia galimybę generuoti tikroviškus vaizdus iš jau turimų duomenų, papildant esamus duomenų rinkinius. Taip gerinamas modelio našumas, ypač palyginus su klasikinėmis augmentacijomis (pasukimais, pakreipimais, ir t.t.), kurios nesukuria visiškai naujų duomenų. Šio tyrimo tikslas – pagerinti saulės kolektorių segmentavimui naudojamo DeepLabV3 modelio našumą naudojant generatyvinį besivaržantį tinklą duomenų augmentacijai. Tyrimo metu DeepLabV3 modeliui naudotas žinių perkėlimas ir modelio adaptavimas o duomenų stokos ir rankinio žymėjimo problema spręsta kuriant naujus duomenis pritaikant pix2pix generatyvinį besivaržantį tinklą.

2 Duomenys ir metodai

Semantinio segmentavimo modelio ir generatyvinio besivaržančio tinklo mokymui naudoti penki skirtingi palydovų nuotraukų su saulės kolektoriais ir jų semantinio segmentavimo kaukių duomenų rinkiniai. Šie rinkiniai yra skirtingų atvaizdo rezoliucijų (1024x1024, 400x400, 256x256), formatų (BMP, PNG) bei erdvinių rezoliucijų (0.8m, 0.3m, 0.2m, 0.1m). Kiekvienos erdvinės rezoliucijos duomenis sudarė 640 nuotraukų ir jų segmentavimo kaukių porų – iš viso 2560 porų. Siekiant išspręsti atvaizdo ir erdvinių rezoliucijų skirtumų problemą bei išvengti skalės neatitikimų, atliktas nuotraukų ir jų kaukių dydžių perskaičiavimas, taikantis į bendrines 512x512 atvaizdo ir 0.1m erdvinę rezoliucijas. Nuotraukos, esančios mažesnės negu nustatyta bendroji erdvinė rezoliucija, padidintos, o tada apkarpytos į 512x512 rezoliuciją, atsižvelgiant į intereso regionus, t.y., saulės kolektorių lokacijas nuotraukose, siekiant išvengti informacijos praradimo. Pix2pix hiperparametrai adaptuoti tinkamam segmentavimo kaukės (tipo A) pavertimui į nuotolinio stebėjimo nuotrauką (tipą B). Segmentavimo modelis mokytas pritaikant žinių perkėlimo techniką, t.y., mokymo metu modelio parametrus atnaujinant pasitelkus anksčiau išmokyto kito modelio žinias, siekiant geresnio modelio tikslumo. Modelis vėliau adaptuotas iš naujo mokant tik paskutinį jo sluoksnį „užšaldant“ kitus sluoksnius, t.y., nekeičiant jų parametrų, sie-

kiant išlaikyti žinių perkėlimo naudą. Tuomet, remiantis paskutinio sluoksnio naujais parametrais, „atšaldyti“ ir mokyti visi likę modelio sluoksniai. Taip segmentavimo modelis tiksliau adaptuotas saulės kolektorių segmentavimo užduočiai.

3 Rezultatai

Modeliai mokyti naudojant 2560 nuotraukų-kaukių poras. 80% duomenų naudota modelio mokymui, o po 10% naudota validavimui ir testavimui. Duomenys, naudoti modelio mokymui, taip pat panaudoti ir pix2pix modelio mokymui – iš tų pačių duomenų sugeneruota 2048 naujų nuotraukų-kaukių porų. DeepLabV3 segmentavimo modelis mokytas šešių eksperimentų metu, sukurti šeši skirtingi modeliai. Modelis No_a mokytas su pradiniu duomenų rinkiniu, o modelis Basic_a mokytas pradiniam duomenų rinkiniui papildomai pritaikant klasikines duomenų augmentacijas. Siekiant patikrinti papildomų duomenų naudą, modelis Gan25 mokytas praplečiant mokymo duomenis panaudojant 25% papildomų sugeneruotų nuotraukų-kaukių porų, o modelis Gan25a mokytas papildomai pritaikant klasikines augmentacijas. Modelis Gan60 mokytas praplečiant duomenų rinkinį optimaliu papildomų sugeneruotų duomenų kiekiu, o modelis Gan60a mokytas pritaikant papildomas klasikines augmentacijas. Atlikus jautrumo analizę, t.y., mokant segmentavimo modelį vis pridedant po 10% sugeneruotų nuotraukų-kaukių porų nustatyta, kad naudojant 60% visų šių naujai sugeneruotų duomenų išgautas didžiausias modelio tikslumas, todėl 1228 nuotraukų-kaukių porų naudota kaip papildomi duomenys jau egzistuojančioms 2048 poroms. Eksperimentų metu mokytų modelių testavimo rezultatai pateikti 1 lentelėje. Lyginant su modeliu No_a, mokytu naudojant pradinį duomenų rinkinį, modelio Gan60 vidutinis pikselių tikslumas (angl. pixel accuracy) padidėjo 0.78%, taiklumas (angl. precision) – 3.41%, jautrumas (angl. sensitivity) – 2.49%, F1 metrika – 2.71%, IoU (angl. intersect over union) metrika – 3.19%, o nuostolių (angl. loss) funkcijos vertė sumažėjo 0.0282. Svarbiausia metrika – IoU, indikuojančios modelio tikslumą aptinkant saulės kolektorius nuotolinio stebėjimo nuotraukose lyginant su segmentavimo kauke. Taip pat padidėjo bendras gerai arba prastai aptiktų saulės kolektorių nuotolinio stebėjimo atvaizduose kiekis, o neaptiktų – sumažėjo.

1 lentelė. Mokytų modelių testavimo rezultatai, pritaikant žinių perkėlimą ir adaptavimą.

Eksperimentas (modelis)	Vid. tiksl. (%)	Vid. taikl. (%)	Vid. jautr. (%)	Vid. F1 (%)	Vid. IoU (%)	Vid. Nuost.	Aptikti saulės kolektoriai (IoU)		
							Gerai ≥ 0.5	Prastai < 0.5	Nėra = 0
No_a	97.89	86.72	85.62	85.25	80.13	0.0650	229	12	15
Basic_a	97.88	89.63	86.51	86.50	81.32	0.0547	235	13	8
Gan25	98.09	89.11	85.94	85.71	80.42	0.0586	229	16	11
Gan25a	97.91	88.84	87.25	87.08	81.41	0.0550	238	8	10
Gan60	98.67	90.13	88.81	87.96	83.32	0.0368	237	11	8
Gan60a	98.04	89.82	87.69	87.77	82.90	0.0611	238	9	9

4 Išvados

Žinių perkėlimas, modelio adaptavimo technikos bei duomenų augmentacijos naudojant generatyvinį besivaržantį tinklą turi, palyginus su klasikinių augmentacijų naudojimu, didesnę naudą semantinio segmentavimo modelio našumui. Padidintas aptiktų saulės kolektorių kiekis bei aptikimo tikslumas, taip pat išvengta rankinio papildomų duomenų žymėjimo turint ribotą kiekį duomenų. Skirtingi duomenų rinkiniai turi būti suvienodinti į bendrą erdvinę rezoliuciją bei vaizdo raišką, siekiant pastovumo modelio mokymo metu, minimalaus informacijos praradimo ir skalės neatitikimo problemos išvengimo. Padidinus mokymo duomenų kiekį naudojant 60% sugeneruotų nuotolinio stebėjimo vaizdų, išmokytas segmentavimo modelis pademonstravo geresnes tikslumo, taiklumo, jautrumo, F1 ir IoU metrikas.

Literatūra

[1] F. M. Guangul, G. T. Chala, “Solar Energy as Renewable Energy Source: SWOT Analysis,” in 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), Jan. 2019, pp. 1–5. doi: 10.1109/ICBDSC.2019.8645580.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation.” arXiv, Mar. 08, 2015. http://arxiv.org/abs/1411.4038

[3] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” arXiv, May 18, 2015. http://arxiv.org/abs/1505.04597

[4] L. Li and E. Lau, “RU-Net: Solar Panel Detection From Remote Sensing Image,” in *2022 IEEE Green Energy and Smart System Systems (IGESSC)*, Long Beach, CA, USA: IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/IGESSC55810.2022.9955325.

[5] F. Ge, G. Wang, G. He, D. Zhou, R. Yin, and L. Tong, "A Hierarchical Information Extraction Method for Large-Scale Centralized Photovoltaic Power Plants Based on Multi-Source Remote Sensing Images," *Remote Sensing*, vol. 14, no. 17, Art. no. 17, Jan. 2022, doi: 10.3390/rs14174211.
[6] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research Progress on Few-Shot Learning for Remote Sensing Image Interpretation," *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, vol. 14, pp. 2387–2402, 2021, doi: 10.1109/JSTARS.2021.3052869.
[7] I. J. Goodfellow *et al.*, "Generative Adversarial Networks." arXiv, Jun. 10, 2014. doi: 10.48550/arXiv.1406.2661.
[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks." arXiv, Nov. 26, 2018. http://arxiv.org/abs/1611.07004

Arrhythmia Classification from ECG Signals Using Transformers and Data Balancing Techniques

Jaunė Malūkaitė, Jolita Bernatavičienė, Povilas Treigys

Vilnius University, Institute of Data Science and Digital Technologies, Akademijos str. 4 Vilnius
jaune.malukaite@mif.stud.vu.lt, jolita.bernataviciene@mif.vu.lt, povilas.treigys@mif.vu.lt

Abstract. While many arrhythmias pose minimal threat, certain heart rhythm irregularities elevate the potential for stroke or heart failure. The complexity arises particularly with the supraventricular premature heartbeat which has a resemblance to a normal beat and occurs infrequently. Consequently, this research proposes a data balancing and classification technique that enhances the accuracy of identifying mentioned hard-to-classify heartbeats while maintaining robust metrics for other classes. The study introduces a deep learning framework combined with a multi-head attention transformer, for balancing – under-sampling and synthetic minority oversampling are used. To evaluate the proposed model, various experiments based on real data were conducted. The results were compared with an existing model used in chest belt heartbeat monitoring, and the results show that the transformer model achieved better performance for supraventricular premature heartbeats, at the same time reaching high overall and per-class metrics.

Keywords: ECG signals, Classification, Deep Learning, Transformer, Focal Loss, Data Balancing Techniques, Heartbeats.

1 Introduction

According to the Lithuanian Institute of Hygiene, more than 22.5 thousand people in Lithuania died in 2022 due to diseases of the circulatory system, accounting for 53 % of all deaths in the country. International data also show that Lithuania's cardiovascular mortality rates are well above the European Union (EU) average and among the highest in the EU. While arrhythmias can be detected from electrocardiograms, the process is time-consuming and prone to errors even among experts. This underscores the significance of automated electrocardiogram analysis. Automated classification of

DOI: https://doi.org/10.15388/LMITT.2024.13

arrhythmias can alleviate the challenging daily workload for medical professionals and facilitate earlier identification of cardiac disorders in patients. Therefore, patients can receive appropriate treatment strategies earlier.

The most common classes of heartbeats analysed in studies are three or four – in this research three classes are chosen, namely supraventricular premature heartbeats (S), normal heartbeats (N) and ventricular premature contraction (V). For the classification of these heartbeats, different machine learning and deep learning models can be used. Approaches such as support vector machine, logistic regression, k-nearest neighbours, and random forest have been used in scientific literature and have demonstrated promising outcomes [3]. However, it has been noted that heart rate classification techniques, which depend on manually extracted features, frequently struggle to discern abstract relationships within the data. Therefore, there is an increasing number of research articles emphasizing the significance of using deep learning methodologies for this purpose [5]. Deep learning transformers have also become increasingly important in recent years, as they have attention mechanisms that give more weight to more important elements in the input sequence. In the arrhythmia classification task, the transformer relies on an attention mechanism and uses the electrocardiogram segments as input to capture global dependencies of signal values [5]. Researchers Rui Hu, Jie Chen, and Li Zhou propose a transformer and neural network architecture wherein a segmented one-dimensional electrocardiogram sequence serves as the input, undergoing multiple one-dimensional convolutional layers. The encoders within the transformer are constructed by iteratively stacking layers with identical structures containing a multi-head self-attention module and a feed-forward network featuring a single hidden layer [2]. The transformer exhibits versatility because it is adaptable not only to convolutional architectures but also to recurrent neural networks. Given the ability of recurrent neural networks to comprehend heart rhythm characteristics, employing a recurrent neural network-based sequence-to-sequence approach could prove advantageous in addressing cardiac classification challenges [4].

The widespread applicability of transformers is evident, hence, in this study, an architecture comprising deep neural networks and transformers is proposed. A significant challenge lies in achieving robust classification

accuracy when using patient data in the test dataset that was not included in the training dataset. Given the anatomical variations among individuals, models tend to emphasize these distinctions over differences in heartbeats themselves. Furthermore, data imbalance poses a recurring complication, as normal heartbeats are disproportionately represented compared to S or V heartbeats, leading to inflated overall metric values primarily driven by the abundance of N class data. Therefore, different data balancing methods are used to overcome this problem [1].

In this research, we seek to improve the supraventricular premature heartbeat classification recall metric by using a transformer model. Furthermore, the impact of different balancing techniques on classification metrics is analysed to find whether data balancing improves metrics. The rest of this paper is organized as follows. In Section 2, information about real data used in the research and its processing is provided. The proposed methodology is discussed in Section 3, while in Section 4 the experimental results are compared and presented. Finally, conclusions are drawn in Section 5.

2 Data

This study utilises data collected from a chest belt for heartbeat monitoring created by Zive company. The dataset consists of 1086 recordings from 102 patients, each lasting 10 minutes. Each recording is subsequently segmented into individual signals. Across the dataset, there are 730860 N heartbeats, 6550 S heartbeats, and 17463 V heartbeats. For additional data combinations and experiments, data from the PhysioNet MIT-BIH Arrhythmia Database is utilised. In the MIT-BIH dataset, which is also used for chest baseline CNN model training, there are 88349 N, 2668 S, and 6783 V heartbeats. While comparing the two datasets, the Zive dataset has more occurrences, especially N heartbeats.

The R peaks of the signals are identified to divide the recordings into heartbeat segments. Following the detection of each R peak, the local minima from the left and the right sides (Q and S' peaks) are found. These peaks create a QRS complex. Additionally, as shown in Figure 1, supplementary parameters such as RRl (the number of signal values to the left closest R peak) and RRr (the number of signal values to the right closest R peak) are computed to ascertain the signal length. The signal is defined as 70 % of RRl values to the left and 70 % of RRr values to the right. A transformation to functional data is then used to standardize all segments to a length of 200.

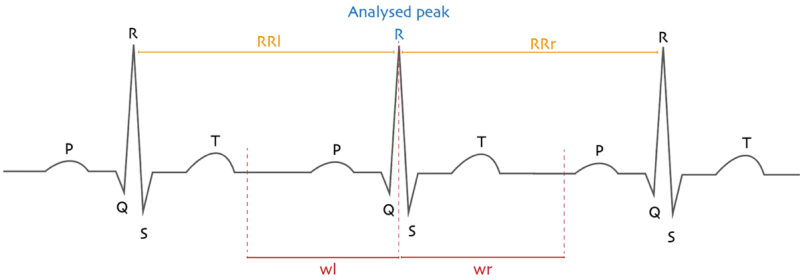


Figure 1. Additional parameters computed from signals. RRl – R peak from the left, RRr – R peak from the right, wl – 70 % of RRl, wr – 70 % of RRr.

Afterwards, data is divided into training, validation, and testing sets. Each set consists of different patient data that were divided into sets by hand to have similar class distributions in all datasets. In the datasets, all 200 signal values, additional derivative features, P, Q, R, S, P values and positions are used.

In the under-sampled dataset, N class occurrences are reduced to 100 thousand while in the SMOTE dataset, N class is reduced to 100 thousand, S class synthetically increased to 20 thousand and V to 40 thousand occurrences. For experimental analysis, different class proportions were used but in further analysis, it was decided to use classes S and V with a balance of 1:2 to have closer to real-life occurrence distribution, in addition to promising first received results. SMOTE numbers are chosen not that large because the synthetical creation can introduce noise and inaccurately imitate heartbeat data. In the original training dataset, there are 380025 N, 4796 S and 11572 V signal segments.

3 Methodology

For data balancing, two different techniques are used: random under-sampling of class N signals and synthetic minority oversampling (SMOTE) [1] of S and V signals in the training set. Random oversampling is not used in this case as the number of S and V classes is very low, and experiments showed that the model tends to learn how to identify only one class with high metrics. SMOTE algorithm creates a new sample for each instance x_i using x_i and its k nearest neighbours in feature space, as defined in the 1

equation. In the equation, x'_i is a new example synthesised from the sample x_i and a randomly selected sample x_j from the nearest neighbours of x_i and λ is a random value from the interval [0, 1] [1].

$$x'_i = x_i + \lambda(x_j - x_i) \quad (1)$$

A custom Focal Loss function is defined and used in the model which is particularly effective for imbalanced datasets, such as those often found in medical diagnosis tasks. This loss function prioritizes challenging instances over simpler ones by adjusting the alpha values used in computations, thereby enhancing focus on harder-to-classify examples. Focal Loss is implemented by adding a modulating factor to the Cross-Entropy loss. In Formula 2, α is considered a weighing factor, p_i is the predicted probability, b is the logarithm base, n is the number of elements being predicted while γ is the focusing parameter. γ rescales the modulating factor such that the easy examples are down-weighted more than the hard ones, reducing their impact on the loss function.

$$Focal\ Loss = -\sum_{i=1}^{i=n} \alpha_i (i - p_i)^{\gamma} \log_b(p_i) \quad (2)$$

The proposed model in the research is created using *PyTorch Lightning* newest version used for streamlined model development and training. The model shown in Figure 2 starts with a multi-headed self-attention mechanism allowing the model to focus on different parts of the ECG signal simultaneously. That is why the input length must be dividable by the number of heads used in the attention mechanism. The mechanism is followed by feed-forward networks within each encoder block. Layer normalization and residual connections stabilize training and facilitate deeper networks. For the transformer encoder, encoder blocks are repeated. Finally, the output from the transformer encoder is passed through a final layer to produce predictions for the ECG signal classes. The decoder part is not used as the encoded signal does not require translation back into a signal for class predictions. The model adopts a 6-layer depth, and a batch size of 128 because it yields better outcomes to 32, 64 or 256. What is more, dropout is integrated to prevent model overfitting, alongside the utilization of a learning rate scheduler and early stopping mechanisms. Optimal epoch checkpoints are stored based on validation loss criteria.

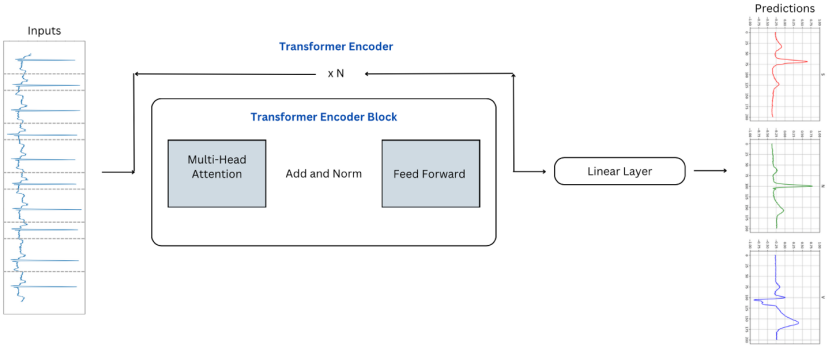


Figure 2. Proposed model architecture scheme that displays the overall model flow and transformer encoder elements.

Base line model is a convolutional neural network with 13 convolutional layers, each followed by batch normalization and activation layers. Batch normalization helps to stabilize and accelerate the training process, while activation layers introduce non-linearity to the network. Additionally, there are two dense layers, a global average pooling layer that helps to reduce the number of parameters. Finally, an output layer is added at the end of the model architecture.

4 Results

For different model and dataset results comparison recall is used as it shows the fraction of instances in a class that the model correctly classified out of all instances in that class. For overall metric calculation macro average recall is chosen to have the classes weighted equally as the amount of N occurrences would distort the results – the weighted metrics would be high even if individual class metrics would be low for S and V classes.

Comparing the model currently used in chest belts and the transformer model with different balance datasets, it is seen from Table 1 that the highest macro average recall for all classes combined is achieved using the transformer model and under-sampled dataset, the metric reaches 0.822 value. Because the research aims to increase class S recall while having N and V class high metrics, the transformer model with an under-sampled dataset achieves 0.570 recall for the S class, which is 0.092 higher than the model in chest belts. N and V class recall values are also high – 0.942 and

0.955 respectively. As for the transformer model with SMOTE dataset, it also achieves promising results, especially for the V class where recall rockets up to 0.983. Regarding the transformer model with original dataset, different parameters and architectures were tried but if one class metrics rise, the other two class metrics decrease.

Table 1. Model recall results using different balance datasets. The model whose results are aimed to be increased is also included for comparison.

Model	N recall	S recall	V recall	Macro avg. recall
Baseline model	**0.997**	0.478	0.934	0.803
Transformer model + under sampled dataset	0.942	**0.570**	0.955	**0.822**
Transformer model + SMOTE dataset	0.929	0.550	**0.983**	0.821
Transformer model + original dataset	0.932	0.559	0.877	0.789

The model used in chest belts has a high weighted average precision of 0.986, recall of 0.987, and f1-score of 0.986 as it learned well class N occurrences and predicts this class most of the time correctly. Because in the test dataset class N appears more often than class S, the overall metrics are high. As for the model with the highest macro average recall score, which is the transformer model used with an under-sampled dataset, the weighted average precision is 0.984, recall 0.939, and f1-score 0.959. In this case, the metrics are also high, while the predictions for the S class have improved.

5 Conclusions

In this paper, we proposed a transformer model which classifies ECG signals into three heartbeat classes. The model architecture and parameters are changed accordingly to experiments made using different balance datasets – under-sampled, oversampled using the SMOTE technique, and the original dataset. In the training, validation, and testing datasets, different patients were used to avoid bias and model learning features relevant to the individuals, not the differences in heartbeats. The best results are received using the proposed transformed model and an under-sampled dataset. The model achieved a macro average recall score of 82.2 %, while the accuracy for the S class, which is the hardest to classify, increased by 9.2 % comparing

the results with a baseline model. In the future, our focus is to increase even more class S metrics by trying out different class proportions in datasets, introducing noisy signals to understand how the model performs when the signals are not high quality, and using transformers together with other deep learning architectures.

6 Acknowledgements

We extend our gratitude to UAB Zive company for the collaboration in data collection, analysis, and comprehension. Additionally, we are thankful for the high-performance computing resources provided by the Information Technology Research Center of Vilnius University.

Research funded under the Programme "University Excellence Initiatives" of the Ministry of Education, Science and Sports of the Republic of Lithuania (Measure No. 12-001-01-01-01 "Improving the Research and Study Environment").

References

[1] J. Chen, J. Lalor, W. Liu, E. Druhl, E. Granillo, V. G. Vimalananda, and H. Yu. Detecting hypoglycemia incidents reported in patients' secure messages: using cost-sensitive learning and oversampling to reduce data imbalance. Journal of medical Internet research, 21(3):e11990, 2019.

[2] R. Hu, J. Chen, and L. Zhou. A transformer-based deep neural network for arrhythmia detection using continuous ecg signals. Computers in Biology and Medicine, 144:105325, 2022.

[3] P. Shimpi, S. Shah, M. Shroff, and A. Godbole. A machine learning approach for the classification of cardiac arrhythmia. In 2017 international conference on computing methodologies and communication (ICCMC), pages 603–607. IEEE, 2017.

[4] B. Wang, C. Liu, C. Hu, X. Liu, and J. Cao. Arrhythmia classification with heartbeat-aware transformer. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1025–1029. IEEE, 2021.

[5] G. Yan, S. Liang, Y. Zhang, and F. Liu. Fusing transformer model with temporal features for ecg heartbeat classification. In 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 898–905. IEEE, 2019.

Extracting TLA$^+$ Specifications Out of a Program for a BEAM Virtual Machine

Andrius Maliuginas, Karolis Petrauskas

Vilnius University, Faculty of Mathematics and Informatics,
Institute of Informatics, Didlaukio g. 47, LT-08303, Vilnius
andrius.maliuginas@mif.stud.vu.lt, karolis.petrauskas@mif.vu.lt

Abstract. Formal specifications are mathematical descriptions of the desired system functionality. Since they are usually written separately from the software itself, it is important to ensure that the software implements what the specification requires. A common approach to achieve this is to have a specification detailed enough to generate source code but those are rarely written due to expertise required. If code is not generated, then currently there is no straightforward way to reliably show that implementation conforms to initial formal specification. This research attempts to define a way to extract formal TLA$^+$ specification by translating Elixir source code and generating detailed specification to give the system developer the ability to show that it refines the initial one.

Keywords: TLA$^+$, Elixir, translation, specification refinement, distributed systems, message passing.

1 Introduction

Distributed systems are well known for their complexity [1]. As a result, many methods have been developed to prevent mistakes during their development, formal specifications being one of them. However, even having a formal specification is not a guarantee of a correct system – there is also a matter of ensuring that implementation conforms to the specification. Since manual analysis is slow and error-prone, several automated methods have been developed over the years to simplify the process, such as generating implementation code from a rather detailed specification (e.g. [2]). In this paper we attempt to go the other way – to develop a method to extract a detailed TLA$^+$ specification from Elixir source code. We do that by defining the Elixir source code translation into TLA$^+$ and generating the detailed specification. Later, refinement mapping could be shown between the generated specification and a more abstract one, thus demonstrating, that implementation has the same properties as the abstract specification.

DOI: https://doi.org/10.15388/LMITT.2024.14

This approach allows avoiding frequent manual changes to detailed specification as the code changes, which results in development process simplification and a decrease in developer expertise required.

We define translation for source code written in the Elixir programming language [3]. It is a language for BEAM virtual machine, which, due to its process model, makes it easy to develop a distributed system. Elixir also has extensive abstract syntax tree manipulation capabilities [4], which helps with source code analysis.

TLA$^+$ [1] has been chosen as a target for our translation. It is a formal specification language, developed to address the challenges posed by specifications of distributed systems. It is a mathematical specifications language, which makes it programming language agnostic and allows specifying systems on a higher level than code.

There have been attempts to develop specification extraction in the past for Erlang programming language, which is another language for BEAM virtual machine, e.g. [5], which translates Erlang into μCRL specification language. We base our work on previous work done for Elixir and TLA$^+$ – [6], which develops a way to translate and generate sequential code into PlusCal and from there into TLA$^+$. In this paper, the focus is on extracting specification for interprocess communication – how messages are sent between the processes. We base our translation on GenServer module usage – it is an Elixir standard library module that simplifies the development of processes that receive messages and keep state [4]. This allows us to look at the system from a higher level and abstract details which are not important for message passing between processes.

In this paper term "translation" refers to the process of turning one language into another, in our case Elixir into TLA+. Term "generation" refers to the automated creation of detailed specification files which contain the translated source code.

2 Distributed systems model

We model a distributed system as a set of processes, which send messages to and receive from a global set of inflight messages. Each process is completely synchronous and independent from others. We consider the set of messages in flight unordered; messages can be delivered to processes in any order. We also assume that processes do not crash, they cannot be created nor destroyed.

We base source code translation on GenServer Elixir module usage, i.e. we consider only implementations that use functions from this module to communicate between the processes. We consider such a decision justified since the GenServer module is a part of the standard library and is commonly used for such tasks.

3 Sequential code translation

Sequential code specification extraction is out of the scope of this investigation. However, we partially define it to the degree that is necessary to extract specification for message passing. Here we present a basic outline of our translation method, albeit incomplete. It is based on an idea developed in earlier work [6].

Since Elixir is a functional programming language, it is convenient to translate sequential code in units of functions. Therefore, each function is expected to be translated into a separate TLA$^+$ module. In Elixir it is possible to give several definitions for the same function, which would be differentiated by passed arguments – during runtime, the first definition, where arguments match parameter types, is executed. In general, it would be more widely applicable to have such pattern matching done inside the function module, however, for our purposes, it was sufficient to treat such definitions as separate functions.

We treat Elixir functions as a series of expressions that are executed one after another. We expect sequential code specification to reflect this – generated specification should consist of a series of operators each of which is a translation of an expression in Elixir. These expressions should be deterministic, that is, given the current process state, they should produce the next process state. For example, given the following Elixir function:

```
def send(n) do
    other_function(n + 1)
end
```

it could be translated as a set of TLA$^+$ operators shown in Listing 1.

$$
\begin{aligned}
&line1(proc) \triangleq \\
&\quad P!\,call(proc, \text{"other\_function"}, \langle P!\,arg(proc, 1) + 1\rangle) \\
&line2(proc) \triangleq \\
&\quad P!\,return\big(proc, P!\,return\_value(proc)\big)
\end{aligned}
$$

Listing 1. Example function expression translations.

In the example above, function body, consisting of a single expression is translated as two separate expressions, *line1* and *line2*. The first one represents the function call together with incrementing its parameter by one while the second one returns the result of the previous function call to the caller. As is evident by this example, not all Elixir expressions are represented as separate expressions in the translation (e.g. parameter increment), nor each operator in translated specification is explicitly reflected in the source code (e.g. function return).

We make use of our Process TLA$^+$ module, which provides operators to access and control the process state. They allow to abstract away the details of common actions away from function modules, making them simpler to translate automatically. Like function expression operators, they are also completely deterministic. This module is included locally in each function module with INSTANCE TLA$^+$ command, as shown in Listing 2. INSTANCE command applied as shown includes all the identifiers of the Process module under the namespace *P*, with Process module constant *Processes* replaced with *Processes* identifier from the current module [7].

$$\text{LOCAL } P \triangleq \text{INSTANCE } Process \text{ WITH } Processes \leftarrow Processes$$

Listing 2. Process module inclusion in function modules.

Function expression operators are meant to be local to the function module. The rest of the generated specification uses *line_enabled* and *line_action* operators. *line_enabled* operator is meant to check if some process is supposed to execute any expression in the current function module. Typically, it should delegate to the Process module operator of the same name. Similarly, the *line_action* operator is given the current process state and a line to execute on that process state and delegates to a correct expression in the module.

Elixir GenServer module function calls are not translated as regular function calls. Instead, we define TLA$^+$ GenServer module, which operators serve as direct equivalents.

4 Specification generation for the entire program

The entire distributed system specification is generated from a template, gaps in which are filled in with source code parts translated into TLA$^+$. This template defines a general execution model for the entire distributed system and handles message deliveries between the processes.

The state of the entire system is split between three variables: *procState*, *sysState*, and *messageQueue*. The last of these, *messageQueue*, is a set of all messages which still have not been received while others store the state of the system itself as it is known for each process. *procState* contains mostly the values used in specification parts that describe the sequential code execution, e.g. function modules. For example, the value of the *procState* variable determines which function expression should be executed on any given process. Meanwhile, *sysState* contains values required for distributed system specification, e.g. what message is currently being processed. *sysState* contains part of the internal GenServer Elixir module functions state. Such separation increases the modularity of the whole method and simplifies the model-checking of any part of sequential code separately from the rest of the generated specification.

Communication between the processes is modelled by a combination of actions, some of which are generated from source code, while others are predefined. Message-receiving actions are generated from GenServer module callback functions handle_cast and handle_call headers. Listing 3 shows how the following GenServer handler function header is translated:

```
def handle_cast({:client, num}, state)
```

The main purpose of the formula in Listing 3 is to match the message in the *messageQueue* and call the respective message handling function with the actual message and current process state as parameters. The actual functionality of the message handler function is to be specified by the function module, the same as for any other sequential code.

$$
\begin{aligned}
&handler1 \triangleq \\
&\quad \exists m \in messageQueue, t \in Processes \\
&\qquad \wedge m.to = t \\
&\qquad \wedge m.msg[1] = \text{"CLIENT"} \\
&\qquad \wedge P!waiting(procState[t]) \\
&\qquad \wedge procState' = upd\_proc\_state(t, \\
&\qquad\quad P!call(\\
&\qquad\qquad P!to\_finished(procState[t]), \\
&\qquad\qquad handle\_cast!name, \\
&\qquad\qquad \langle m.msg,\ sysState[t].state \rangle)) \\
&\qquad \wedge messageQueue' = M!drop(messageQueue, m) \\
&\qquad \wedge sysState' = upd\_sys\_state\big(t, S!set\_reply\_to(sysState[t], m)\big) \\
&\qquad \wedge \text{UNCHANGED } nextMsgId
\end{aligned}
$$

Listing 3. Message receiving action example.

Other actions related to message passing are there to ensure the system state is updated as expected after the received message is handled and to take care of synchronous communication. *handler_finished* action does both jobs simultaneously – it updates the system state after the handler finishes and sends out the response message, which may be returned by the handler function. The other two actions, *waiting_responses* and *deliver_responses* are there to correctly translate GenServer multicall function call which sends the same message to several recipients and waits for their responses. We do not provide definitions for these actions here due to space constraints; definitions can be found in the code repository[1].

Sequential code execution is specified by *function_lines* action. It is defined as a disjunction of formulas of the structure shown in Listing 4. The entire disjunction is also existentially quantified to select any process, which allows to model-check different expression execution orderings for a group of processes. *fn_line* operator is displayed in Listing 5. If some function expression can be executed, it updates the process state, sends out all produced messages and starts waiting for replies to the synchronous messages sent.

$$
\begin{array}{l}
\exists l \in function!lines: \\
\quad \text{LET} \\
\quad\quad line\_enabled \triangleq function!line\_enabled(procState[p], \\
\quad\quad line\_result \triangleq function!line\_action(procState[p], l) \\
\quad \text{IN} \\
\quad\quad fn\_line(p, line\_enabled, line\_result)
\end{array}
$$

Listing 4. Structure of function module expression execution block.

$$
\begin{array}{l}
fn\_line(process, line\_enabled, line\_result) \triangleq \\
\quad \text{LET} \\
\quad\quad becomes\_blocked \triangleq P!blocked(line\_result) \\
\quad\quad complete\_messages \triangleq M!full\_msgs(line\_result.sent\_msgs) \\
\quad \text{IN} \\
\quad\quad \wedge\, line\_enabled \\
\quad\quad \wedge\, procState' = upd\_proc\_state(process, line\_result) \\
\quad\quad \wedge\, messageQueue' = M!bulk\_send(messageQueue, complete\_messages) \\
\quad\quad \wedge\, nextMsgId' = nextMsgId + Cardinality(complete\_messages) \\
\quad\quad \wedge\, \text{IF } becomes\_blocked \text{ THEN} \\
\quad\quad\quad sysState' = set\_wait\_replies\_for(process, complete\_messages) \\
\quad\quad \text{ELSE} \\
\quad\quad\quad \text{UNCHANGED } sysState
\end{array}
$$

Listing 5. *fn_line* operator definition.

[1] https://github.com/mr-frying-pan/master

In all listings provided in this section, we use operators from modules referred to as *M* and *S*. These names stand for Messaging and System TLA$^+$ modules, respectively. Similarly to the Process module described in Section 3 these modules provide operators for their respective areas – message passing and system state modifications. They are included in the specification in the same way as the Process module – using INSTANCE command.

5 Work in progress

Experiment to verify the applicability of the developed method to a realistic algorithm is currently in progress. We have generated a specification for our implementation of Bracha reliable broadcast [8]. We attempt to show that the generated specification is a refinement of an abstract Bracha reliable broadcast specification. Abstract specification of Bracha reliable broadcast, our Elixir implementation and generated specification for it are available in the source code repository[2].

Message-passing part of the specification was generated according to the proposed method. To perform model-checking, sequential code specification is also needed. Since sequential code generation is outside the scope of this investigation, it was written manually. Despite that, manually written function modules retain the required operators so that they can be used in generated specification with minimal changes to it.

We try to show the refinement with model-checking, by showing that abstract specification holds as a property when model-checking generated specification. So far, an initial refinement mapping has been defined; however, the correctness of the mapping is yet to be shown, and we continue tuning the refinement.

6 Conclusions

The developed translation method is modular, different modules encapsulate their respective areas well. If necessary, it is possible to prove properties for any module separately, for both predefined modules and

[2] Repository can be found in https://github.com/mr-frying-pan/master.
Abstract specification is in gen_spec/tla/BrachaRBC.tla.
Our Elixir implementation is in bracha/lib/bracha.ex.
Main generated specification file is gen_spec/tla/bracha.tla.

sequential code modules. We attempt to limit the state explosion by having completely deterministic operators where it is possible to have them, making the number of states dependent on initial inputs.

Future work in the area is needed to further limit state explosion since currently there are a lot of orderings sequential expressions could be executed in, in addition to the message delivery orderings.

Also, more work is needed to obtain a fully functional specification generator. Currently, we generate only overall specification, without the function modules while the bulk of functionality for some algorithm often would be implemented as sequential operations. The sequential code generator is currently being developed and will have to be incorporated into the existing one once it is finished.

Synchronous communication between processes also requires future work, especially the specification of timeouts. It is possible to add timeouts into our specification, but it would require handling process failures and errors.

References

[1] L. Lamport, J. Matthews, M. Tuttle and Y. Yu, "Specifying and verifying systems with TLA+," in *Proceedings of the 10th workshop on ACM SIGOPS European workshop*, 2002.

[2] J. R. Wilcox, D. Woos, P. Panchekha, Z. Tatlock, X. Wang, M. D. Ernst and T. Anderson, "Verdi: A Framework for Implementing and formally verifying distributed systems," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2015.

[3] S. Juric, Elixir in Action, Manning, 2019.

[4] D. Thomas, Programming Elixir ≥ 1.6: Functional |> Concurrent |> Pragmatic |> Fun, Pragmatic Bookshelf,, 2018.

[5] T. Arts, C. B. Earle and J. J. S. Penas, "Translating Erlang to μCRL," in *Proceedings. Fourth International Conference on Application of Concurrency to System Design, 2004. ACSD 2004*, 2004.

[6] D. Bražėnas, *Extracting TLA+ Specifications out of Elixir Programs,* Vilnius: Vilnius University, 2023.

[7] L. Lamport, Specifying systems: the TLA+ language and tools for hardware and software engineers, Addison-Wesley, 2002.

[8] G. Bracha, "Asynchronous Byzantine agreement protocols," *Information and Computation,* vol. 2, no. 75, p. 130–143, 1987.

Draudimo sektoriaus klientų atsiliepimų ir vertinimų nuotaikų kaitos analizė laike

Donata Petkutė, Gražina Korvel

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Duomenų mokslo ir skaitmeninių technologijų institutas,
Akademijos g. 4, Vilnius
donata.petkute@mif.stud.vu.lt

Santrauka. Šiandien internetas tampa nepakeičiamas informacijos šaltinis, kuriame gausu įvairių atsiliepimų apie įsigytus produktus ar paslaugas. Šie atsiliepimai teikia vertingą informaciją įmonėms, norinčioms geriau suprasti savo klientų poreikius ir lūkesčius. Vienas iš efektyviausių būdų išgauti įžvalgas iš atsiliepimų yra naudoti nuotaikų analizę. Šiame tyrime aptariama, kiek klientų yra patenkinti ir nepatenkinti draudimo sektoriaus teikiamomis paslaugomis bei siūlomais produktais, taip pat nuotaikų priskyrimui buvo naudojami skirtingi vektorizavimo ir klasifikavimo metodai, kad būtų pasiekti geriausi rezultatai. Analizei atlikti naudojami du duomenų rinkiniai – produkto įsigijimo ir atsiliepimai apie žalos išmokėjimą po draudiminio įvykio. Tyrime naudojami du vektorizavimo būdai – žodžių maišo ir TF-IDF bei trys klasifikavimo metodai: atraminių vektorių, naiviojo Bajeso bei ilgalaikės trumposios atminties modelis. Atlikus tyrimą gauta, jog klientų atsiliepimų nuotaikas geriausiai klasifikuoja naiviojo Bajeso klasifikatorius su TF-IDF vektorizavimo būdu, kai tikslumas siekia 91% abiem duomenų rinkiniams. Atsiliepimams po produkto įsigijimo gautos preciziškumo ir atkūrimo metrikos teigiamam sentimentui 93% ir 97% atitinkamai, neigiamai klasei 73% ir 55%. Teigiamai klasei po žalų atlyginimo gautas preciziškumas 93% ir atkūrimo metrika 96%, o neigiamai – 82% ir 72% . Pritaikius atraminių vektorių klasifikatorių su skirtingomis vektorizavimo technikomis gauta tikslumo įvertis 89%.

Raktiniai žodžiai: nuotaikų analizė, natūralios kalbos apdorojimas, mašininis mokymasis, TF-IDF, žodžių maišas.

1 Įvadas

Šiais laikais socialinė žiniasklaida atlieka labai svarbų vaidmenį beveik kiekvieno žmogaus kasdieniame gyvenime, pavyzdžiui, suteikia vartotojams galimybę išreikšti savo nuomonę apie tam tikrą produktą ar paslaugą [6, 10]. Iš tikrųjų vis dažniau žmonės pasikliauna kitų klientų patirtimi ir naudojasi atsiliepimų apžvalgomis, kurios padeda nuspręsti produkto įsigijimo svar-

DOI: https://doi.org/10.15388/LMITT.2024.15

bą. Vieni žmonės produktui skiria keturis ar penkis balus ir išreiškia galutinį pasitenkinimą produktu, o kiti skiria vieną ar du − išreikšdami visišką nepasitenkinimą. Tai nekelia jokių sunkumų siekiant suprasti klientų nuotaiką. Tačiau kiti žmonės skiria tris balus, nors akivaizdžiai išreiškia visišką pasitenkinimą produktu. Tai klaidina kitus klientus, taip pat įmones, norinčias sužinoti tikrąją nuomonę [2]. Įmonėms tampa būtina suprasti žmonių nuotaikas, tam kad galėtų ištirti vartotojų nuomonę ir požiūrį į jų paslaugas bei atrastų naujų verslo strategijų [3, 12]. Siekiant geriau suvokti bendrą vartotojų požiūrį ir pasitenkinimą teikiamomis paslaugomis, verta atlikti nuotaikų analizę, kurios metu bandoma nustatyti ar sakinys yra teigiamas, ar neigiamas. [10, 11]. Dažnai produktai neatitinka klientų lūkesčių, todėl nuotaikų analizės naudojimas po naujo produkto išleidimo į rinką, bendrovėms gali padėti suprasti jo trūkumus ir pranašumus [3]. Tačiau visuomenės nuomonė apie tam tikrą temą laikui bėgant kinta, todėl siekiant nustatyti tendencijas bei sezoniškumą svarbu atlikti analizę laike. Be to, tokia analizė padeda nustatyti nukrypimus, kurie gali būti susiję su įvykiais, sukėlusiais nuotaikų pokyčius [4].

M. F. Madjid ir kt. atliko nuotaikos analizę, nagrinėdami programų atsiliepimus, naudodami atraminių vektorių modelį (angl. Support vector machine, SVM) ir naiviojo Bajeso (angl. Naive Bayes, NB) klasifikatorių [16]. Tyrime gauti rezultatai rodo, kad atraminių vektorių metodo tikslumas pasiekia 94,29 %, o Naiviojo Bajeso klasifikatorius – 93,97 % tikslumą. Tuomet atliekant oro linijų atsiliepimų nuotaikų analizę A. M. Rohat ir kt. naudojo naiviojo Bajeso ir atraminių vektorių modelį. Rezultatai taip pat parodė, kad oro linijų atsiliepimų atveju SVM užtikrina daug geresnius tikslumo rezultatus (82 %), o NB algoritmas – tik 76 % [14]. Kitą sentimentų klasifikavimo analizę atliko J.J. A. Limbong ir kiti tyrėjai, kuriuo metu buvo naudojami naiviojo Bajeso ir k artimiausių kaimynų (angl. k-nearest neighbors, KNN) klasifikatoriai, rezultatai parodė, kad KNN metodas veikia geriau, jo klasifikavimo tikslumo vertė yra 92,8 %, palyginti su NB metodo tikslumo verte – 91,4 % [15]. Taip pat pastaraisiais metais gilusis mokymasis sulaukia vis daugiau dėmesio pramonėje ir akademiniame pasaulyje dėl savo didelio našumo įvairiose srityse. Šiuo metu populiariausi giliojo mokymosi architektūros tipai yra pasikartojantis neuroninis tinklas (angl. Reccurent Neural Network, RNN) ir konvoliucinis neuroninis tinklas (angl. Convolutional Neural Network, CNN) [7]. Straipsnyje [11] tyrėjai taikė gilaus mokymosi metodus − konvoliucinį neuroninį tinklą, ilgosios trumpalaikės atminties (angl. Long Short-Term

Memory, LSTM) modelį bei paprastąjį neuroninį tinklą (angl. Simple Neural Network), kad atliktų Twitter nuotaikų analizę. Autoriai gavo, jog LSTM yra geriausias iš visų naudotų metodų, jo tikslumas yra 87 %, o CNN ir paprastojo neuroninio tinklo metodai atitinkamai pasiekė 82 % ir 81 % tikslumą. Kiekvienu atveju norint įvertinti klasifikavimo rezultatus buvo naudojamos tikslumo metrikos: tikslumas, preciziškumas, atkūrimas, F1 statistika.

2 Pirminis teksto apdorojimas

Pirminis teksto apdorojimas taikomas siekiant išvalyti ir paruošti tekstą nuotaikų klasifikavimui, nes dažnu atveju vartotojų rašomi tekstai yra nestruktūrizuoti. Tokiuose tekstuose paprastai yra daug nereikalingos, nenaudingos informacijos, pavyzdžiui, pasikartojančių žodžių, skaičių, skyrybos ženklų, rašybos klaidų, jaustukų ir sutrumpinimų. Darbe pirmiausia atsiliepimai buvo suskaidyti į teksto vienetus. Toliau pašalinami nereikšmingi žodžiai, tai gali būti jungtukai, įvardžiai, kurie laikomi nereikalingais ir nenaudingais.

3 Teksto vektorizavimas

Atlikus pirminį teksto apdorojimą, kitas žingsnis pritaikyti vektorizavimą, kuris tekstinius duomenis paverčia skaitiniu vektoriumi. Darbe taikomi du skirtingi vektorizavimo būdai – žodžių maišo ir TF-IDF. Kadangi dauguma mašininio mokymosi algoritmų duomenis apdoroja su skaitiniais įvesties reikšmėmis, tai yra būtinas žingsnis atliekant nuotaikų analizę.

Žodžių maišas (angl. Bag-of-words, BOW) − tai natūraliosios kalbos apdorojimo metodas, naudojamas tekstiniam dokumentui atvaizduoti kaip žodžių rinkiniui, neatsižvelgiant į jų pateikimo tvarką. Žodžių maišo metodas yra vienas paprasčiausių tokio tipo metodų, skaičiavimo bei konceptualumo prasme. Pagrindinė idėja – suskaičiuoti kiekvieno žodžio dažnumą dokumente ir remiantis šiais žodžių dažniais, sukurti dokumento vektorinį atvaizdavimą [30].

Termino dažnis-atvirkštinio dokumento dažnis (angl. Term Frequency-Inverse Document Frequency, TF-IDF) populiarus tyrimų metodas natūralios kalbos apdorojimo srityje. TF-IDF metodu nustatomas santykinis žodžių dažnis konkrečiame dokumente. Žodžiai, kurie tekste pasitaiko dažniau yra laikomi mažiau svarbiais, o rečiau pasitaikantys žodžiai priskiriami svarbesniems, nes laikoma, kad jie turi daugiau reikšmingos informacijos [1]. Metodo matematinė išraiška parodyta žemiau esančioje lygtyje:

$$\log(1 + tf_{t,d}) \times idf_t = log_{10} \frac{N}{df_t}$$

čia *N* nurodo dokumentų numerį rinkinyje, $tf_{t,d}$ kaip dažnai žodis *t* pasitaiko dokumente *d*, o idf_t apibūdina paieškos termino svarbą visų kolekcijos dokumentų atžvilgiu.

4 Temų modeliavimas

Temų modeliavimas − tai natūralios kalbos apdorojimo uždavinys, kai duomenų taškai grupuojami į klasterį atsižvelgiant į jų panašumą, o tie, kurie neturi panašumų, bus sugrupuoti į kitus klasterius. Temų modeliavimas apibrėžiamas kaip būdas sugrupuoti duomenis į klasterius taip, kad tame pačiame klasteryje esantys duomenys turėtų daugiau panašumų, palyginti su skirtinguose klasteriuose esančiais duomenimis [8].

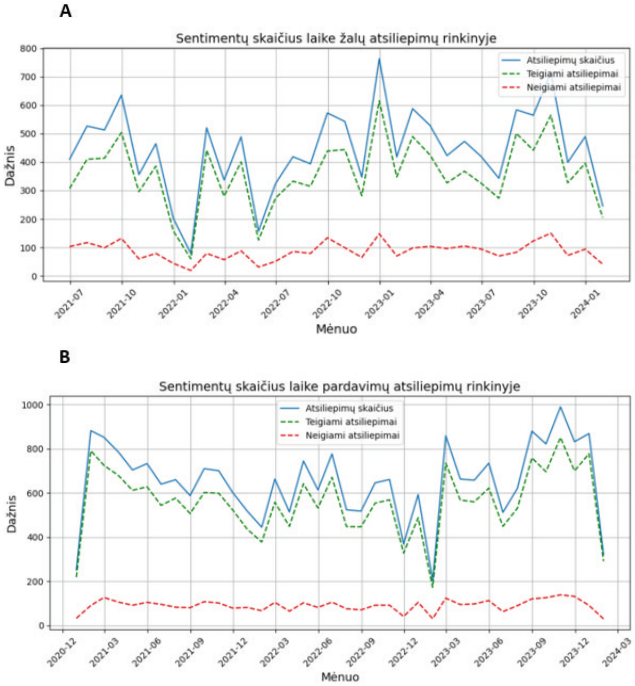
LDA (angl. Latent Dirichlet allocation) yra generuojantis tikimybinis modelis, kurį 2003 m. pirmą kartą pristatė H. Jelodar ir kt [5]. Pagrindinė idėja yra ta, kad dokumentai pateikiami kaip latentinių temų atsitiktiniai mišiniai, kur temą apibūdina žodžių pasiskirstymas. LDA pateikia temas pagal žodžių tikimybes.

5 Eksperimento rezultatai

Norint įvertinti skirtingus vektorizavimo ir klasifikavimo metodus, buvo pasirinkta analizuoti du duomenų rinkinius – atsiliepimai įsigyjant produktą ir klientų vertinimai po draudiminio įvykio atlyginimo. Atlikus pirminį teksto apdorojimą pardavimų atsiliepimų rinkinyje iš viso liko 24662 įrašai, o antrajame duomenų rinkinyje – 14248 atsiliepimai, kurie buvo pateikti klientų po žalos atlyginimo. Abu duomenų rinkiniai turi stulpelį su įvertinimais, pagal kuriuos atsiliepimai buvo suskirstyti į dvi grupes: teigiamus ir neigiamus. Gauta, jog pirmajame duomenų rinkinyje yra 21219 teigiami ir 3443 neigiami atsiliepimai, o antrame rinkinyje – 11478 teigiami ir 2770 neigiami atsiliepimai.

1 paveiksle parodytas bendras, teigiamų ir neigiamų atsiliepimų, paskelbtų kiekvieną mėnesį skirtinguose duomenų rinkiniuose, skaičius. Iš A grafiko, kuris vaizduoja atsiliepimus po žalų atlyginimo matosi, jog didėjant atsiliepimų skaičiui didėja ir neigiami atsiliepimai, tačiau galime pastebėti, jog nuo 2023 metų vasario mėnesio, kai parašomų vertinimų skaičius krito,

o neigiamų atsiliepimų kiekis laikėsi tolygiai. Vertinant abu grafikus pastebima, jog daugiausia klientai vertinimus palieka spalio mėnesiais. Didžiausi kiekiai atsiliepimų A grafiko atveju 2023 sausio mėnesį, o B atveju 2023 metų lapkričio mėn.



1 pav. Žalų atlyginimo (A) ir pardavimų (B) bendrų, neigiamų ir teigiamų atsiliepimų skaičius kas mėnesį.

Neigiamų atsiliepimų temos buvo modeliuojamos kiekvienam ketvirčiui. Norėdami įvertinti sugeneruotų temų kokybę buvo taikomas nuoseklumo (angl. Coherence) balas, kuris padeda nustatyti ar temos yra aiškios ir gerai apibrėžtos. Šis balas remiasi sąsaja tarp žodžių temoje. Pirmiausia, išrenkami svarbiausiai žodžiai kiekvienai temai, o tada parenkamos visos galimos žodžių poros iš atrinktų žodžių. Toliau skaičiuojamas panašumas kiekvienai žodžių porai, matuojant, kaip dažnai du žodžiai pasirodo kartu. Visi panašumai susumuojami kiekvienai temai, gaunant bendrą temos vertę. Galiausiai apskaičiuojamas nuoseklumo balas, dažniausiai kaip vidurkis arba media-

na iš visų temos verčių. Kuo aukštesnis balas, tuo labiau temos apibrėžtos. Gauta, jog didžiausią nuoseklumo balą turinčių temų skaičius yra 10 pardavimų atsiliepimams. 1 lentelėje pateiktos sumodeliuotos temos su raktiniais žodžiais 2023 metų IV ketvirčiui, pardavimų rinkinio atsiliepimams. Iš gautų temų, galima matyti, jog klientai labiausiai nepasitenkinę kainomis, informacijos trūkumu, tikisi geresnio pasiūlymo.

1 lentelė. Pardavimų atsiliepimo rinkinio 2023 metų IV ketvirčiui temų raktiniai žodžiai.

Tema	Raktiniai žodžiai
1	Mažas, draudimas, kainuoti, kaina
2	Pinigai, galėti, kainas, pigiai
3	Galėti, nuolaida, pasiūlymas, draudimas
4	Sunkus, klausimas, įvykis, draudimas
5	Darbuotojas, produktas, turtas, draudimas
6	Paslauga, kaina, šalis, draudimas
7	Darbuotojas, informacija, nežinoti, draudimas
8	Geresnis, bendravimas, kaina, klientas
9	Balas, didelis, draudimas, kaina
10	Bendravimas, žala, trūkti, draudimas

Analogiškas temų modeliavimas atliktas ir antram duomenų rinkiniui, kur didžiausią nuoseklumo balą turinčių temų skaičius yra 3 neigiamų atsiliepimų atveju. Iš 2 lentelėje esančių rezultatų matome, kad daugiausiai tarp temų pasikartoja žodis *draudimas*, *žala*, galime daryti išvadas, kad klientai labiausiai nepatenkinti žalos išmokėjimu, taipogi susidūrė su problemomis naudojantis pakaitiniu automobiliu.

2 lentelė. Atsiliepimų po žalų atlyginimo sumodeliuotų 2023 metų IV ketvirčiui temų raktiniai žodžiai.

Tema	Raktiniai žodžiai
1	Įvykis, draudimas, pakaitinis, automobilis
2	Klientas, žala, kaina, draudimas
3	Žala, darbuotojas, trūkti, draudimas

Sekantis žingsnis - klasifikavimo etapas, naudojant dvi skirtingas vektorizavimo technikas (žodžių maišo ir TF-IDF) bei taikant atraminių vektorių ir naiviojo Bajeso klasifikavimo metodus. Šie klasifikavimo metodus pasirinkti remdamiesi mokslinės literatūros analize ir atliktais tyrimais. Geriausiems rezultatams pasiekti buvo naudojama parametrų gardelė. Kiekvieno modelio rezultatai pateikti 3 ir 4 lentelėse, iš kurių matyti, jog naudojant TF-IDF kartu su naiviuoju Bajeso klasifikatoriumi pasiekiami didžiausi tikslumai abiem duomenų rinkiniams. Mažiausi įverčiai gauti naudojant žodžių atraminių vektorių klasifikatorių.

3 lentelė. Pardavimų atsiliepimų rinkinio klasifikavimo tikslumo metrikos.

Klasifikavimo metodas	Vektorizavimo metodas	Sentimentas	Preciziškumas (%)	Atkūrimas (%)	F1 metrika (%)	Bendras tikslumas
SVM	TF-IDF	Neigiamas	69	45	54	89%
		Teigiamas	91	97	94	
	BOW	Neigiamas	68	44	53	89%
		Teigiamas	91	97	94	
NB	TF-IDF	Neigiamas	73	55	63	91%
		Teigiamas	93	97	95	
	BOW	Neigiamas	66	55	60	90%
		Teigiamas	93	95	94	

4 lentelė. Atsiliepimų rinkinio po žalų atlyginimo klasifikavimo tikslumo metrikos.

Klasifikavimo metodas	Vektorizavimo metodas	Sentimentas	Preciziškumas (%)	Atkūrimas (%)	F1 metrika (%)	Bendras tikslumas
SVM	TF-IDF	Neigiamas	78	63	70	89%
		Teigiamas	91	96	93	
	BOW	Neigiamas	79	60	68	89%
		Teigiamas	90	96	93	
NB	TF-IDF	Neigiamas	82	72	77	91%
		Teigiamas	93	96	95	
	BOW	Neigiamas	76	64	69	89%
		Teigiamas	91	95	93	

6 Išvados

Šiame tyrime buvo naudojami du skirtingi metodai vektorizavimui ir klasifikavimui, siekiant nustatyti klientų atsiliepimų nuotaikas. Taikytos TF-IDF ir žodžių maišo vektorizavimo technikos, kartu su naiviuoju Bajeso ir atraminių vektorių algoritmais. Geriausi rezultatai gauti naudojant naiviojo Bajeso klasifikatorių kartu su TF-IDF, kuris pasiekė net 91% tikslumą skirtingiems duomenų rinkiniams. Atraminių vektorių klasifikatorius pasiekė 89% tikslumą pirmam ir antram duomenų rinkiniams. Lyginant preciziškumo ir atkūrimo metrikas gauta, jog geriausiai kiekvieną klasę atskiria naudojant naiviojo Bajeso kartu su TF-IDF vektorizavimo metodu. Atsiliepimams po produkto įsigijimo geriausiai atskiriama teigiama klasė, tada preciziškumas lygus 93%, o atkūrimo metrika 97%. Atsiliepimams po žalų atlyginimo teigiami sentimentai taip pat geriau atskiriami, gaunamas preciziškumas 93% ir atkūrimo metrika 96%. Nors buvo išbandyti tik keli algoritmai, tolimesniems tyrimams būtų tikslinga išbandyti kitus algoritmus arba kurti hibridinius metodus, siekiant padidinti rezultatų tikslumą. Taip pat atliekant tyrimą pastebėta, kad duomenys nėra balansuoti, todėl ateities darbas bus subalansuoti duomenis. Klientų atsiliepimų nuotaikų nustatymas gali būti naudingas įvairiose srityse. Galimos išmaniosios sistemos, kurios galėtų pateikti vartotojams išsamias produktų, paslaugų ir kt. apžvalgas, nereikalaujant, kad vartotojai peržiūrėtų atskiras apžvalgas, o galėtų tiesiogiai priimtų sprendimus remdamiesi sistemos pateiktais rezultatais.

Literatūra

[1] W. N. I. Al-Obaydy, H. A. Hashim, Y. Najm, and A. A. Jalal. Document classification using term frequency-inverse document frequency and k-means clustering. Indonesian Journal of Electrical Engineering and Computer Science, 27(3):1517–1524, 2022.

[2] A. S. AlQahtani. Product sentiment analysis for amazon reviews. International Journal of Computer Science & Information Technology (IJCSIT) Vol, 13, 2021.

[3] C. Chauhan and S. Sehgal. Sentiment analysis on product reviews. In 2017 International Conference on Computing, Communication and Automation (ICCCA), pages 26–31. IEEE, 2017.

[4] A. Giachanou and F. Crestani. Tracking sentiment by time series analysis. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pages 1037–1040, 2016.

[5] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. Multimedia tools and applications, 78:15169–15211, 2019.

[6] K. S. Kumar, J. Desai, and J. Majumdar. Opinion mining and sentiment analysis on online customer review. In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pages 1–4. IEEE, 2016.
[7] G. Murthy, S. R. Allu, B. Andhavarapu, M. Bagadi, and M. Belusonti. Text based sentiment analysis using lstm. Int. J. Eng. Res. Tech. Res, 9(05), 2020.
[8] E. S. Negara, D. Triadi, and R. Andryani. Topic modelling twitter data with latent dirichlet allocation method. In 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), pages 386–390. IEEE, 2019.
[9] A. W. Sari, T. I. Hermanto, and M. Defriani. Sentiment analysis of tourist reviews using k-nearest neighbors algorithm and support vector machine. Sinkron: jurnal dan penelitian teknik informatika, 8(3):1366–1378, 2023.
[10] T. Shivaprasad and J. Shetty. Sentiment analysis of product reviews: A review. In 2017 International conference on inventive communication and computational technologies (ICICCT), pages 298–301. IEEE, 2017.
[11] A. C. M. V. Srinivas, C. Satyanarayana, C. Divakar, and K. P. Sirisha. Sentiment analysis using neural network and lstm. In IOP conference series: materials science and engineering, volume 1074, page 012007. IOP Publishing, 2021.
[12] A. Tripathy, A. Agrawal, and S. K. Rath. Classification of sentimental reviews using machine learning techniques. Procedia Computer Science, 57:821–829, 2015.
[13] Rahat, A. M., Kahir, A., & Masum, A. K. M. (2019, November). Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset. In 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART) (pp. 266-270). IEEE.
[14] Limbong, J. J. A. (2022). Analisis Klasifikasi Sentimen Ulasan Pada E-Commerce Shopee Berbasis Word Cloud Dengan Metode Naive Bayes Dan K-Nearest Neighbor (Doctoral dissertation).
[15] Y. Zhang, R. Jin, and Z.-H. Zhou. Understanding bag-of-words model: a statistical framework. International journal of machine learning and cybernetics, 1:43–52, 2010.
[16] Madjid, M. F., Ratnawati, D. E., & Rahayudi, B. (2023). Sentiment Analysis on App Reviews Using Support Vector Machine and Naïve Bayes Classification. Sinkron: jurnal dan penelitian teknik informatika, 8(1), 556-562.

Vaizdų aprašų generavimo modeliai

Artūr Radzivilov

Vilniaus Gedimino technikos universitetas,
Saulėtekio al. 11, LT-10223 Vilnius
artur.radzivilov@vilniustech.lt

Santrauka. Šiame straipsnyje yra nagrinėjami vaizdų aprašų generavimo modeliai, kurių pagalba galima automatizuoti teksto aprašymų kūrimą iš vaizdinės informacijos. Pateikiamos įvairios neuroninių tinklų struktūros, tokios kaip CNN ir RNN, kurios naudojamos vaizdų savybių išgavimui ir teksto generavimui, bei dėmesio mechanizmai ir „transformer" tipo tinklai, leidžiantys geriau integruoti vaizdo ir tekstinę informaciją. Analizuojami pagrindiniai duomenų rinkiniai, naudojami modelių mokymui, ir aprašymų vertinimo metodai, skirti įvertinti generuotų teksto aprašymų kokybę. Taip pat aptariamos naujausios tendencijos ir iššūkiai šioje srityje, pabrėžiant būsimų tyrimų kryptis.

Raktiniai žodžiai: vaizdų aprašų generavimas, CNN, RNN, dėmesio mechanizmai.

1 Įvadas

Šiame apžvalginiame straipsnyje nagrinėjami šiuolaikiniai vaizdų aprašymo generavimo modeliai, siekiant išanalizuoti kaip šie modeliai leidžia automatizuoti vaizdų aprašymus. Straipsnis skirtas išsamiai apžvelgti dabartinėje mokslinėje literatūroje aprašytas metodikas ir duomenų rinkinius, kuriuos naudoja vaizdų aprašymo sistemų kūrėjai. Pabrėžiama neuroninių tinklų, tokių kaip konvoliuciniai (CNN) ir rekurentiniai (RNN) neuroniniai tinklai, naudojimo svarba, taip pat aptariami dėmesio mechanizmai ir transformer tipo tinklai, kurie atlieka esminį vaidmenį siekiant efektyviai integruoti vizualinę ir tekstinę informaciją. Be to, šiame straipsnyje pateikiama vertinimo metodų analizė, leidžianti įvertinti generuotų aprašymų kokybę, ir aptariamos įvairios esamos technologijos bei jų stiprybės ir silpnybės [1].

2 Duomenų rinkiniai

Duomenų rinkiniai yra labai svarbus dalykas vaizdų aprašų generavimo sistemoje. Tam, kad sistema galėtų parodyti rezultatą, palyginamą su žmogaus vaizdo aprašymo rezultatu, reikalingi labai dideli duomenų masyvai,

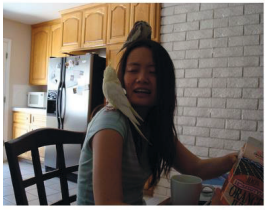
DOI: https://doi.org/10.15388/LMITT.2024.16

kuriuose privalo būti vaizdai ir nors vienas aprašas atitinkantis vaizdui. Kuo daugiau vienas vaizdas turi aprašų, tuo geriau, nes tą patį vaizdą skirtingi žmonės gali aprašyti skirtingai. Apmokant vaizdų aprašų generavimo sistemą, kuri galėtų pakeisti žmogų, reikia, kad aprašai duomenų rinkiniuose būtų sukurti žmogaus ranka.

Duomenų rinkinis „PASCAL1K“ buvo sukurtas 2010 metais. Jis turi 9000 vaizdų ir daugiau nei 40 000 aprašų šitiems vaizdams. „Amazon‘s Mechanical Turk“ (MTurk) padėjo autoriams sukurti šį duomenų rinkinį. MTurk leidžia pakankamai greitai rinkti didelį kiekį lingvistinių duomenų, nereikalaujant santykinai didelių investicijų.

MTurk taip pat turi ir minusų, vienas iš jų tai ribota galimybė kontroliuoti tai, kas iš personų gali dalyvauti tam tikroje užduotyje. Atliekant užduotis, reikalaujančias teksto įvedimo laisva forma, tai tas minusas, dėl kurio gali iškilti problemos. Tokio tipo užduotys skiriasi nuo užduočių su keliais galimais atsakymo variantais, kurie dažniausiai būna sukurti testine forma. Tai reiškia, kad tokio tipo užduočių rezultatą negalima bus patikrinti naudojant testinę užduotį, atsakymas į kurią yra žinomas. Dar vienas minusas susijęs su tuo, kad MTurk neturi įrankio, kuris garantuotų, kad visos personos, kurios spręs užduotis, gerai žinos anglų kalbą.

Kvalifikaciniai testai yra procedūros, kurias atlieka asmenys, norintys dalyvauti duomenų rinkimo procese, siekiant įsitikinti, kad jie atitinka tam tikrus kvalifikacijos standartus, pavyzdžiui, kalbos mokėjimą arba gebėjimą teisingai atlikti užduotis, prieš jiems leidžiant dalyvauti tekstų generavimo ir kitose panašiose užduotyse.



Without qualification test
(1) lady with birds
(2) Some parrots are have speaking skill.
(3) A lady in their dining table with birds on her shoulder and head.
(4) Asian woman with two cockatiels, on shoulder
head, room with oak cabinets.,
(5) The lady loves the parrot
With qualification test
(1) A woman has a bird on her shoulder, and another bird on her head
(2) A woman with a bird on her head and a bird on her shoulder.
(3) A women sitting at a dining table with two small birds sitting on her.
(4) A young Asian woman sitting at a kitchen
table with a bird on her head and another on her shoulder.
(5) Two birds are perched on a woman sitting in a kitchen.

1 pav. Aprašymų su ir be kvalifikacinio testo pavyzdys „PASCAL1K“ duomenų rinkinyje [2].

Kaip galima pamatyti 1 pavyzdyje su aprašais, rezultatai po kvalifikacinių testų ir be jų labai skiriasi. Kvalifikaciniai testai padėjo išvengti punktuacijos, rašybos ir kitų klaidų. Pasirinkti konkretesni aprašymai be tariamų santykių tarp objektų, kaip, pavyzdžiui, 5 punkte: „The lady loves the parrot“ (Ponia myli papūgą).

„Flickr8K“ ir „Flickr30K“ duomenų rinkiniai, sukurti nuotraukų aprašymų generavimo modelių mokymui, sudaryti iš tūkstančių įvairių tematikų nuotraukų, kurioms būdingi detalūs aprašymai, padedantys mokymosi procese [3]. „Flickr30K Entities“ papildo šiuos rinkinius, įtraukiant objektų lokalizaciją ir žodžių susiejimą su vaizdais, taip pagerinant modelių sugebėjimus generuoti tikslų ir kontekstinį turinį. Klaidos rinkinyje rodo, jog yra tobulėjimo galimybių, ypač kalbant apie rėmelių tikslumą ir objektų identifikaciją. „Microsoft COCO Captions“ duomenų rinkinys, praturtintas gausiais vaizdais ir aprašymais iš įvairių kategorijų, teikia dar didesnę įvairovę modelių mokymui, orientuotą į objektų atpažinimą, lokalizavimą ir aprašymų generavimą. Šie duomenų rinkiniai atlieka svarbų vaidmenį tobulinant ir vertinant vaizdų aprašymų generavimo technologijas, suteikdami tyrėjams reikiamus įrankius aiškesnėms ir tikslesnėms vizualizacijoms kurti.

3 Sugeneruotų aprašų vertinimas

Automatinių vaizdų aprašymų generavimo sistemos vertinimas yra sudėtingas procesas, kuris reikalauja objektyvių metrikų, kad būtų galima išmatuoti sugeneruotų aprašymų kokybę. Šiam tikslui mokslininkai ir inžinieriai naudoja įvairias vertinimo metrikas, tokias kaip BLEU, ROUGE, METEOR, CIDEr ir SPICE, kurios leidžia vertinti aprašymus remiantis skirtingais aspektais, įskaitant gramatiką, turinio tikslumą, sinonimų naudojimą ir semantinę prasmę. Kiekviena iš šių metrikų turi savo stiprybes ir silpnąsias puses, todėl jų kombinavimas suteikia išsamesnį ir objektyvesnį modelių vertinimą [4].

Metrikos, kaip BLEU ir ROUGE, dažnai naudojamos vertinant tekstų sutapimą ir aprašymų išsamumą [5], o METEOR prideda papildomą sluoksnį, atsižvelgdamas į gramatiką ir sinonimus [6]. Tuo tarpu CIDEr ir SPICE yra orientuoti į semantinį turinį ir aprašymų atitikimą žmogaus sukuriam konsensusui [7]. Visos šios metrikos padeda identifikuoti, kiek gerai automatinių sistemų generuoti aprašymai atitinka realius vaizdus ir jų kontekstą, tačiau taip pat pabrėžia, kad nėra vienos universalios vertinimo sistemos.

Būsimų tyrimų kryptys apima tiek esamų metrikų tobulinimą, tiek naujų metodų kūrimą, siekiant dar tiksliau įvertinti aprašymų kokybę ir atspindėti sudėtingesnius modelių generavimo aspektus. Svarbu atsižvelgti į modelių sudėtingumą, apmokymo laiką ir efektyvumą, kad būtų galima išsamiai įvertinti jų privalumus ir trūkumus. Tokie vertinimai yra būtini ne tik mokslinės pažangos požiūriu, bet ir praktine prasme, siekiant sukurti efektyvesnes ir energiją taupančias modelių architektūras, kurios atveria naujas taikymo sritis ir pagerina technologijų naudojimą kasdieniame gyvenime.

4 Egzistuojančios vaizdų aprašų generavimo sistemos

Vaizdų aprašymų generavimo sistemos yra svarbus dirbtinio intelekto ir kompiuterinės regos tyrimų segmentas, kuris siekia sukurti modelius, galinčius analizuoti vaizdus ir generuoti apie juos natūralios kalbos aprašymus. Šios sistemos remiasi pažangiomis dirbtinio intelekto technologijomis, tokiomis kaip konvoliuciniai neuroniniai tinklai (CNN), vaizdų savybių išgavimui, ir rekurentiniai neuroniniai tinklai (RNN), tekstui generuoti, taip pat dėmesio mechanizmais ir „transformer“ tipo tinklais, siekiant efektyviai apdoroti ir integruoti vaizdo bei tekstinius duomenis.

NIC yra vienas iš pirmųjų metodų šioje srityje, kuris efektyviai derina CNN, vaizdų savybių išgavimui, ir RNN, aprašymų generavimui, demonstruodamas kaip šie du komponentai gali bendradarbiauti generuojant aprašymus. NIC modelis, kuriame CNN naudojamas kaip koduotojas ir RNN kaip dekoduotojas, yra fundamentali koncepcija, kuri buvo pritaikyta ir tobulinta įvairiose vėlesnėse sistemose [8].

„Show, Attend and Tell“ metodas įvedė dėmesio mechanizmą į vaizdų aprašymų generavimo procesą, leidžiant sistemai ne tik identifikuoti vaizde esančius objektus, bet ir nustatyti kuriuos objektus ir kada reikėtų akcentuoti generuojant kiekvieną aprašymo žodį. Tai suteikė modeliams galimybę generuoti žymiai tikslesnius ir konteksto atžvilgiu prasmingesnius aprašymus [9].

„Transformer“ tipo tinklai, su jų naujovišku dėmesio mechanizmu, leidžia efektyviai tvarkyti ir analizuoti didelius duomenų kiekius, taip pat pritaikyti modelius įvairioms NLP užduotims, įskaitant vaizdų aprašymų generavimą. Jų gebėjimas apdoroti visą įvestį vienu metu, o ne paeiliui reiškia didžiulį žingsnį efektyvumo ir veikimo supratimo požiūriu [10].

„Image Transformer“ pritaiko „transformer“ tipo tinklų architektūrą tiesiogiai vaizdams, naudodamas lokalų dėmesį ir kvadratinį kontekstą vaizdo

pikselių analizei. Ši metodika leidžia modeliui efektyviai ir tikslingai generuoti vaizdo aprašymus, koncentruojantis į svarbiausius vaizdo elementus [11].

VLP (Vision and Language Pre-training) metodai, tokie kaip „ViLBERT“ ir „VisualBERT“, rodo, kad išankstinis mokymas, naudojant tiek vaizdo, tiek teksto duomenis, gali suteikti modeliams reikiamą lankstumą ir pritaikymą įvairioms vaizdo ir kalbos sąveikos užduotims atlikti. Šie metodai, derinantys galingas „transformer“ tipo tinklų architektūras su išankstiniu mokymu, atveria naujas galimybes modelių veikimo gerinimui vaizdų aprašymų generavimo srityje.

Šių sistemų plėtra ir tobulinimas leidžia ne tik generuoti tikslius ir prasmingus vaizdų aprašymus, bet ir gilinti mūsų supratimą apie vaizdo ir teksto sąveiką. Toliau vykstantys tyrimai ir technologinė pažanga tikrai atneš dar daugiau inovacijų ir tobulinimų, leidžiančių dar labiau pagerinti šių sistemų efektyvumą ir universalumą.

5 Neuroninių tinklų struktūros

Neuroninių tinklų struktūros yra esminė dirbtinio intelekto technologijų dalis, lemianti jų efektyvumą ir taikymą plačiame spektre užduočių. Šios struktūros atlieka svarbų vaidmenį analizuojant vaizdus, generuojant tekstus, atpažįstant kalbą ir sprendžiant kitus uždavinius.

CNN yra pagrindiniai vaizdo apdorojimo ir analizės darbuose, įskaitant objektų atpažinimą, vaizdų klasifikavimą ir segmentavimą. Jų struktūra, sudaryta iš konvoliucinių, aktyvavimo ir slenkstinių sluoksnių, leidžia efektyviai išmokti vaizdų savybes nuo paprastų iki sudėtingų. CNN naudojimas yra itin veiksmingas dėl gebėjimo atpažinti ir išgauti svarbius vaizdo bruožus, pritaikant mažiau parametrų ir išteklių nei kitos architektūros [12].

RNN yra skirti dirbti su sekos duomenimis, tokiomis kaip tekstas ar garso įrašai. Jų pagrindinė savybė – gebėjimas išsaugoti informaciją per ilgesnį laikotarpį. RNN architektūra leidžia informacijai keliauti nuo vieno apdorojimo žingsnio prie kito. Šis „žingsnis“ reiškia vieną iteraciją per duomenų seką, kurioje tinklas apdoroja vieną elementą (pvz., žodį ar garsą) ir perduoda savo būseną į kitą iteraciją. Tokiu būdu modelis įgauna gebėjimą atskleisti ir išmokti priklausomybes tarp iš eilės einančių sekos elementų, kas yra ypač svarbu užduotims kuriose reikia suprasti ir prognozuoti sekos tęstinumą.

Pagrindinės RNN sudedamosios dalys yra vidinė būsena, kuri atnaujinama kiekviename žingsnyje perduodant informaciją, ir grįžtamoji ryšio

struktūra, leidžianti informacijai cirkuliuoti per tinklą [13]. Tačiau RNN dažnai susiduria su gradientų nykimo problema, kuri apsunkina ilgalaikių priklausomybių mokymąsi [14].

Dėmesio mechanizmai padidina neuroninių tinklų efektyvumą, leisdami modeliams sutelkti dėmesį į svarbiausius duomenų elementus konkrečiu metu [9]. Tai yra ypač naudinga ilgų sekų apdorojimui, pavyzdžiui, teksto vertimui ar vaizdų analizei, nes dėmesio mechanizmai dinamiškai supranta, kiekvieno įvesties ar sekos elemento svarbą [15].

Architektūra „transformer" tipo tinklų yra pažangi, remiasi dėmesio mechanizmais ir atsisako tradicinių CNN ar RNN elementų. Ši architektūra yra labai veiksminga NLP užduotims ir vaizdų analizei dėl savo gebėjimo efektyviai apdoroti ilgas sekas, nenaudojant brangių rekursijos ar konvoliucijos operacijų. „Transformer" tipo tinklai yra pagrindas šiuolaikiniams AI modeliams, tokiems kaip GPT ir BERT, suteikiantys jiems gebėjimą suprasti ir generuoti kalbą bei analizuoti vaizdus nepaprastai aukštu lygiu.

„Encoder-decoder" architektūra yra būdinga užduotims, kuriose koduotojas apdoroja įvestį, o dekoduotojas generuoja išvestį. Ši schema naudojama įvairiose užduotyse, pavyzdžiui, mašininiame vertime, teksto generavime ir vaizdų aprašymų kūrime [16]. „Encoder-decoder" modeliai gali būti pagrįsti RNN, CNN ar „transformer" tipo tinklų architektūra ir dažnai įtraukia dėmesio mechanizmus, kad padidintų jų veiksmingumą ir gebėjimą išmokti sudėtingas priklausomybes tarp įvesties ir išvesties duomenų [10].

Šios pagrindinės neuroninių tinklų architektūros atlieka lemiamą vaidmenį formuojant šiuolaikinio dirbtinio intelekto technologijas, suteikdamos mokslininkams ir inžinieriams įrankius sudėtingų problemų sprendimui ir naujų, pažangių sistemų kūrimui.

6 Išvados

Šiame straipsnyje apžvelgti vaizdų aprašymų generavimo modeliai, remiantis naujausiomis dirbtinio intelekto technologijomis, tokiomis kaip CNN, RNN, dėmesio mechanizmai ir „transformer" tipo tinklai, parodo, kaip šios technologijos gali efektyviai transformuoti vizualinę informaciją į tekstinius aprašymus. Sistemos naudoja įvairius duomenų rinkinius modelių mokymui ir tobulinimui, o sugeneruotų aprašymų vertinimas atliekamas naudojant išsamią metrikų analizę, tokią kaip BLEU, ROUGE ir kitos, leidžiančią išsamiai įvertinti aprašymų kokybę ir nustatyti tobulinimo kryptis.

Tęsiant mokslinius tyrimus ir technologijų plėtrą, galime tikėtis, kad vaizdų aprašymų generavimo technologijos toliau evoliucionuos, tobulindamos jų sugebėjimą suprasti ir interpretuoti vizualinius bei tekstinius duomenis. Ateityje šios technologijos gali transformuoti tai, kaip mes suvokiame ir naudojame vaizdinę informaciją, integruodamos jas į įvairias pramonės šakas ir kasdienę veiklą, tokiu būdu suteikdamos naujas galimybes vartotojų sąveikai su technologijomis.

Literatūra

[1] Koršunova KP. Tasks and methods of automatic image description. Systems of Control, Communication and Security 2018;1:30–77.

[2] Rashtchian C, Young P, Hodosh M, Hockenmaier J. Collecting Image Annotations Using Amazon's Mechanical Turk 2010:139–47.

[3] Hodosh M, Young P, Hockenmaier J. Framing Image Description as a Ranking Task Data, Models and Evaluation Metrics Extended Abstract. IJCAI International Joint Conference on Artificial Intelligence 2015:4188–92.

[4] Kuznetsova P, Ordonez V, Berg AC, Berg TL, Choi Y. Collective Generation of Natural Image Descriptions 2012:8–14.

[5] Papineni K, Roukos S, Ward T, Zhu W-J. Bleu: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02 2002:311–8. https://doi.org/10.3115/1073083.1073135.

[6] Denkowski M, Lavie A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. Proceedings of the Ninth Workshop on Statistical Machine Translation, Stroudsburg, PA, USA: Association for Computational Linguistics; 2014, p. 376–80. https://doi.org/10.3115/v1/W14-3348.

[7] Vedantam R, Zitnick CL, Parikh D. CIDEr: Consensus-based image description evaluation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE; 2015, p. 4566–75. https://doi.org/10.1109/CVPR.2015.7299087.

[8] Vinyals O, Toshev A, Bengio S, Erhan D. Show and Tell: A Neural Image Caption Generator. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2014;07-12-June-2015:3156–64. https://doi.org/10.1109/CVPR.2015.7298935.

[9] Xu K, Ba JL, Kiros R, Cho K, Courville A, Salakhutdinov R, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 32nd International Conference on Machine Learning, ICML 2015 2015;3:2048–57.

[10] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. Adv Neural Inf Process Syst 2017;2017-December:5999–6009.

[11] Parmar N, Vaswani A, Uszkoreit J, Kaiser L, Shazeer N, Ku A, et al. Image Transformer. 35th International Conference on Machine Learning, ICML 2018 2018;9:6453–62.

[12] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition 2015.

[13] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks. Adv Neural Inf Process Syst 2014;4:3104–12.

[14] Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation 2014.
[15] Luong M-T, Pham H, Manning CD. Effective Approaches to Attention-based Neural Machine Translation 2015.
[16] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate 2014.

Macroeconomic Influences on Baltic Housing Loan Flows

Mija Aneta Stasiulionytė

Vilnius University, Faculty of Mathematics and Informatics,
Naugarduko g. 24, Vilnius, Lithuania
mija.stasiulionyte@mif.stud.vu.lt

Abstract. This study investigates the impact of macroeconomic factors on housing loan flows in the Baltic region post-2008 housing bubble. Using stepwise regression, Lasso regression, and pooled regression, data is analyzed from Estonia, Latvia, and Lithuania spanning 2014 to 2023. Results reveal the significant influence of borrowing costs, wages, unemployment, and inflation rates on loan flows.

Keywords: housing loans, macroeconomics, LASSO regression, linear regression, pooled regression, Baltic region.

1 Introduction

Despite studies on credit and housing in Eastern Europe [1] , research specifically focusing on the influence of macroeconomic factors on housing loan flows in the Baltics after the housing bubble burst in 2008 remains relatively understudied. This study addresses this gap by examining loan data in the Baltics. Two approaches are employed: analyzing loan flows in each Baltic state (Estonia, Latvia, Lithuania) and using pooled regression analysis across region. This study aims to provide a comprehensive understanding of how macroeconomic factors spanning from 2014 to 2023, and for Lithuania from 2015 to 2023, influence housing loan flows patterns in the Baltics.

2 Methods & Results

In this study, stepwise regression [2] with the “both” method was used to construct a regression model of loan flows for house purchases based on macroeconomic factors. Stepwise regression was chosen for its systematic approach to predictor selection, balancing model complexity and goodness of fit using the Akaike Information Criterion (AIC). This method selects

DOI: https://doi.org/10.15388/LMITT.2024.17

predictors, starting with an empty model, and sequentially adding and removing predictors based on their statistical significance until no further improvement in model fit is observed. AIC guided model selection, with lower AIC values indicating a better fit. The resulting model coefficients, see Table 1, highlight the significant impacts of borrowing costs (*Borr_cost_house*) and wages across countries. Higher borrowing costs were associated with reduced loan accessibility for households, while increasing wages positively influenced loan accessibility. Inflation (*HICP*) positively impacts housing loan flows in Lithuania and Estonia, while unemployment (*Unemploy*) negatively affects loan flows in Lithuania.

Least Absolute Shrinkage and Selection Operator (LASSO) regression [3] aims to minimize prediction error by constraining variables and coefficients. LASSO regression, adept at handling multicollinearity, aided in identifying significant predictors while ensuring model simplicity through coefficient shrinkage. The selection of the shrinkage parameter (λ) was performed using k-fold cross-validation. Results showed in Table 2, it is observed that the coefficients related to private consumption *(Priv_consum*) were shrunk to zero in the Lasso regression analysis, indicating their lack of statistical significance, consistent with the findings from linear regression and pooled regression for a region. However, borrowing costs (*Borr_cost_house*) and wages variables retained their coefficients in the Lasso regression, suggesting their consistent and significant impact on loan flows for house purchases. Additionally, Lasso regression highlighted the influence of unemployment (*Unemploy*) solely in Lithuania, with no effect observed in other countries.

The pooled regression model was utilized to offer a thorough perspective on the Baltic region. This method treated all observations equally, assuming homogeneity across both countries and time periods. Analysis (Table 1) reveals that borrowing costs (*Borr_cost_house*) and wages exhibit consistent and significant impacts across all regions, mirroring their individual effects observed at the country level. Similarly, unemployment (*Unemploy*) and inflation rates (*HICP*) emerge as significant factors influencing the entire region, despite their prior examination being confined to specific countries. A noteworthy regional finding is the discernible influence of government consumption (Gov_consum) on housing loan flows, this insight was exclusively revealed by the Lasso regression and was not evident in the linear regression.

Table 1. Regression results.

	Dependent variable:			
	Housepurch_l			
	Lithuania	Estonia	Latvia	Pooled Regression
	(1)	(2)	(3)	
Unemploy	-2.862**			-9.015***
	(1.396)			(0.855)
HICP	0.896***	1.513***		0.787***
	(0.305)	(0.282)		(0.242)
log(Wages)	65.965***	77.887***	52.755***	36.564***
	(5.804)	(8.950)	(3.959)	(3.636)
Borr_cost_house	-13.429***	-12.424***	-6.924***	-8.814***
	(1.691)	(2.013)	(1.066)	(1.526)
Gov_consum				-10.405***
				(2.162)
OutlierLt	-130.660***			
	(13.238)			
OutlierLv		110.752***		
		(8.432)		
Constant	-376.857***	-448.368***	-337.136***	-127.422***
	(42.470)	(56.692)	(26.261)	(25.886)
AIC	540.83	630.9	500.01	1041.21
Observations	105	117	117	339
R2	0.831	0.687	0.781	0.613
Adjusted R2	0.823	0.679	0.775	0.607
Residual Std. Error	12.836 (df = 99)	14.620 (df = 113)	8.331 (df = 113)	21.306 (df = 333)
F Statistic	97.617***(df = 5; 99)	82.861***(df = 3; 113)	134.165***(df = 3; 113)	105.303(df = 5; 333)
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 2. LASSO coefficients.

	Lithuania	Estonia	Latvia
Unemploy	-1.504	0	0
HICP	0.887	1.623	0
log(Wages)	66.385	68.512	52.319
Priv_consum	0	0	0
Invest	0.836	-0.111	-0.039
Gov_consum	5.635	3.898	-0.208
Borr_cost_house	-12.993	-9.690	-6.607
OutlierLt	-132.471	0	0
OutlierLv	0	0	108.203
Constant	-392.580	-396.241	-335.134

Research [4] supports the notion that as interest rates increase, housing affordability declines. Conversely, increasing wages positively impact loan accessibility, as highlighted by [5] article, which suggests that rising wages

contribute to greater household wealth, thereby stimulating demand for loans, including those for house purchases, and indicating improved well-being. Article [6] suggests that inflation has an impact on loan flows as borrowers seek loans to mitigate the negative effects of inflation on their purchasing power. Study [7], which focuses on the US, offers relevant insights indicating that unemployment can reduce disposable income and increase credit risk for borrowers. This diminishes their attractiveness as loan candidates, resulting in negative impacts on loan applications.

3 Conclusion

Through a synthesis of empirical evidence and existing literature this study contributes to the understanding of housing loan dynamics in the Baltic region following the housing bubble burst. By employing stepwise regression, LASSO regression, and pooled regression techniques, it is systematically analyzed the influence of macroeconomic factors on loan flows. Findings underscore the significance of borrowing costs, wages, unemployment, and inflation rates in shaping loan accessibility across the region.

References

[1] Walko, Z. (2008). Housing Loan Developments in the central and eastern European EU member States. Focus on European economic integration, 2(08), 73-82.

[2] Hwang, J. S., & Hu, T. H. (2015). A stepwise regression algorithm for high-dimensional variable selection. Journal of Statistical Computation and Simulation, 85(9), 1793-1806.

[3] Ranstam, J., & Cook, J. A. (2018). LASSO regression. Journal of British Surgery, 105(10), 1348-1348

[4] Chaney, A., & Hoesli, M. (2010). The interest rate sensitivity of real estate. Journal of Property Research, 27(1), 61-85.

[5] Zezza, G. (2008). US growth, the housing market, and the distribution of income. Journal of Post Keynesian Economics, 30(3), 375-401.

[6] Alter, M. A., & Mahoney, E. M. (2020). Household debt and house prices-at-risk: A tale of two countries. International Monetary Fund.

[7] Donaldson, J. R., Piacentino, G., & Thakor, A. (2019). Household debt overhang and unemployment. *The Journal of Finance*, *74*(3), 1473-1502.

[8] Murauskas, G., & Čekanavičius, V. (2014). Taikomoji regresinė analizė socialiniuose tyrimuose. Vilniaus universiteto leidykla.

Skaitmeninio dvynio koncepcinė analizė

Martynas Valatka

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Didlaukio g. 47, LT-08303 Vilnius
martynas.valatka@mif.stud.vu.lt

Santrauka. Skaitmeniniai dvyniai yra naudojami labai įvairiose tiek mokslo, tiek pramonės srityse ir reiškia objektų, sistemų ar procesų virtualių kopijų kūrimą. Tačiau nėra vieningos skaitmeninio dvynio sampratos. Mokslinėje literatūroje skaitmeniniais dvyniais vadinamas platus modelių spektras – nuo naudotojo profilio iki realios esybės išsamaus vykdomo analogo. Šio darbo tikslas yra atlikus sistemingą literatūros analizę atsakyti į klausimą „Kokias būtinas savybes turi turėti esybės modelis, kad jį būtų galima pavadinti skaitmeniniu dvyniu?“. Atsakant į tyrimo klausimą sudarytas skaitmeninio dvynio metasavybių rinkinys, kuris nusako minimalią savybių grupę, kai skaitmeninis dvynys yra suprantamas kaip virtuali realios esybės kopija.

Raktiniai žodžiai: skaitmeninis dvynys, koncepcinė analizė, skaitmeninis modelis, metasavybė, sisteminga literatūros apžvalga.

1 Įvadas

Skaitmeniniai dvyniai yra gana plati samprata naudojama įvairiose srityse, susijusi su objektų, sistemų ar procesų virtualių kopijų kūrimu. Šios skaitmeninės kopijos palengvina sistemų veikimo stebėjimą, analizę ir optimizavimą realiu laiku, gali suteikti gilių įžvalgų apie sistemos elgseną ir našumą.

Akademinėje bendruomenėje nėra vieningos skaitmeninio dvynio sampratos. Skaitmeninių dvynių pritaikymas apima didžiulį spektrą sričių, todėl egzistuoja daug skirtingų jų apibrėžčių bei jiems keliami labai skirtingi reikalavimai [1, 2, 3]. Darbe [4] pateikta išsami skaitmeninių dvynių apibrėžčių apžvalga, tačiau nustatytos savybės leidžia suprasti skaitmeninius dvynius realybės atspindžio, replikos prasme. O tai reiškia maksimalią galimų savybių aibę. Todėl yra aktualu nustatyti, kokias būtinas savybes turi turėti skaitmeniniai dvyniai, identifikuoti jų tipus. Šio darbo tikslas ir yra atsakyti į klausimą „Kokias būtinas savybes turi turėti realios esybės modelis, kad jį būtų galima pavadinti skaitmeniniu dvyniu?“.

DOI: https://doi.org/10.15388/LMITT.2024.18

Straipsnyje pateikiami sistemingos literatūros analizės rezultatai. Ją atlikus nustatyta, kad nėra vieningos nuomonės apie būtinas ir pakankamas skaitmeninio modelio savybes. Sudarytas skaitmeninio dvynio metasavybių rinkinys, kuris nusako minimalią savybių grupę, kai skaitmeninis dvynys yra suprantamas kaip virtuali realios esybės kopija.

2 Tyrimo atlikimo metodika

Siekiant atsakyti į tyrimo klausimą atlikta sisteminga literatūros apžvalga, taikant PRISMA [5] metodiką. Rezultatai gauti keturiais etapais. Pirmame etape buvo nustatyti straipsniai, kuriuos reikia peržiūrėti. Antrame ir trečiame etape buvo atlikta straipsnių peržiūra ir priimtas sprendimas dėl jų tinkamumo išsamiai analizei. Paskutiniame etape buvo atlikta koncepcinė analizė, kurios rezultate sudarytas skaitmeninio dvynio metasavybių rinkinys.

Organizacinis atliktos sistemingos literatūros analizės aspektas, nusakantis paieškos vykdymo detales, pateiktas 1 lentelėje.

1 lentelė. Literatūros paieškos proceso charakteristika

Paieškos tikslas	Skirtingų skaitmeninių dvynių apibrėžčių apžvalga
Raktiniai žodžiai	Digital twin, digital twin definition, digital twin concept, digital twin model, digital twin characteristic
Šaltinių atrankos kriterijai	Publikacijos, randamos pasauliniame saityne
Kalba	Anglų kalba
Šaltiniai	Google Scholar, IEEE library, ACM Digital Library, DBLP
Šaltinių tipai	Žurnalų publikacijos, konferencijų pranešimų medžiaga, daktaro disertacijos, techninės ataskaitos
Pradinio publikacijų sąrašo atrankos kriterijai	Pavadinimas, santrauka, raktiniai žodžiai, įvadas

3 Skaitmeninio dvynio sampratos analizė

3.1 Skaitmeninio dvynio samprata

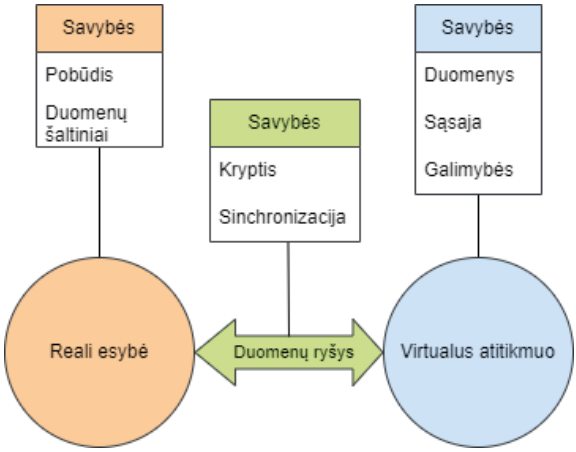
Nagrinėjant platų straipsnių apie skaitmeninius dvynius spektrą, apimantį įvairias mokslo ir pramonės sritis, pastebėta, jog skaitmeniniai dvyniai apibūdinami skirtingai, išryškinamos skirtingos skaitmeninių dvynių savybės bei jiems keliami labai skirtingi reikalavimai. Nors vieningo skaitmeninių dvynių apibrėžimo nėra, tačiau bendru atveju visi literatūroje pateikiami

skaitmeninių dvynių apibrėžimai iš esmės atitinka bendrą sampratą, kai skaitmeniniai dvyniai suvokiami, kaip virtualios realios esybės kopijos, taigi, turi būti neatsiejamai nagrinėtinos trys pagrindinės dalys:

1. reali esybė realioje erdvėje,
2. virtualus esybės atitikmuo virtualioje erdvėje,
3. duomenų bei informacijos srautai tarp realios ir virtualios esybių.

3.2 Skaitmeninio dvynio metasavybės

Analizuotų šaltinių pateikiamose sampratose išryškėjo 7 pagrindinės skaitmeninio dvynio metasavybės susijusios su kiekviena iš esminių jo dalių (1 pav.). Metasavybė – tai esybę kaip tipą nusakanti charakteristika.



1 pav. Skaitmeninių dvynių metasavybės

Realią esybę, visų pirma, apibrėžia jos pobūdis. Dažniausiai literatūros šaltiniuose kalbama apie fizinių sistemų skaitmeninius dvynius [6], tačiau kiti autoriai pažymi, jog virtualiais modeliais vaizduoti galima ir įvairius procesus, programų sistemas ar bet kokius kitus nematerialius konceptus [7]. Taip pat, pagal realios esybės pobūdį, skaitmeniniai dvyniai gali būti klasifikuojami į komponentų, produktų, sistemų ir procesų dvynius [8]; atitinkamai pagal tai ar modeliu vaizduojama tik nedidelė realios esybės dalis, ar kelių dalių, produktų ar sistemų tarpusavio sąveika. Duomenų šaltinių savybė paprasčiausiai nurodo, kokie sensoriai ar kitos duomenų rinkimo sistemos naudojamos duomenims apie realią esybę išgauti.

Pagrindinė savybė charakterizuojanti virtualų atitikmenį yra jį sudarantys duomenys. Pasak vienų autorių, virtualus modelis privalo atvaizduoti visus įmanomus išgauti duomenis apie realią esybę, jos elgseną bei aplinką [9], tačiau kiti autoriai mano, jog tokio lygmens detalumas yra sunkiai pasiekiamas ir daugeliu paktinio pritaikymo atvejų užtenka atsirinkti tik pakankamus duomenis skaitmeniniam dvyniui iškeltiems tikslams pasiekti. Pagal detalumą skaitmeninius dvynius galima klasifikuoti į selektyvius (*ang. selective*) ir visapusiškus (*ang. comprehensive*). Literatūros šaltinių pateikiamose sampratose taip pat daug dėmesio skiriama skaitmeninių dvynių teikiamoms galimybėms. Dažniausiai minimos stebėjimo [10], aptikimo [10], prognozavimo [11] ir simuliacijos [12] galimybės. Kadangi skaitmeniniai dvyniai neprivalo įgyvendinti visų šių galimybių, juos galima klasifikuoti į aprašomuosius, informatyviuosius, prognozuojamuosius, išsamiuosius ir autonominius dvynius [8], atitinkamai pagal teikiamų galimybių kiekį ir sudėtingumą. Galiausiai, sąsajos savybė apibrėžia, kaip suteikiama prieiga prie skaitmeninio dvynio teikiamų duomenų vartotojui, kitoms mašinoms ar net kitiems skaitmeniniams dvyniams [13].

Duomenų ryšį charakterizuoja krypties bei sinchronizacijos savybės. Šios savybės sukelia daugiausiai išsiskiriančių nuomonių, kadangi, pasak vienų autorių, skaitmeninis dvynys privalo duomenis atsinaujinti realiu laiku, ne tik pasyviai gauti duomenis, bet ir aktyviai veikti savo realų atitikmenį [9], tačiau kiti autoriai argumentuoja, jog, priklausomai nuo nagrinėjamos esybės bei išsikeltų reikalavimų dvyniui, gali pakakti periodiško duomenų atnaujinimo ir vienpusio duomenų sąryšio [7]. Pagal šias savybes skaitmeninius dvynius galima klasifikuoti į vienakrypčius ir dvikrypčius bei realaus ar beveik realaus laiko ir periodiškai atsinaujinančius dvynius.

4 Išvados

Mokslinėje literatūroje skaitmeniniais dvyniais vadinamas platus modelių spektras – nuo naudotojo profilio iki realios esybės išsamaus vykdomo analogo. Būtinų skaitmeninio dvynio savybių identifikavimą apsunkina ir labai skirtingas jų klasifikavimas: pagal funkcionalumą, struktūriniu požiūriu, galimybę imituoti arba simuliuoti, naudojimo pobūdį, adekvatumą modeliuojamai esybei. Tačiau nežiūrint į minėtus skirtingus aspektus, bet kurį skaitmeninį dvynį sudaro trys būtinos dalys. Todėl skaitmeninio dvynio metasavy-

bių rinkinys, kuris nusako minimalią savybių grupę, kai skaitmeninis dvynys yra suprantamas kaip virtuali realios esybės kopija, yra šis: realios esybės pobūdis ir esminiai duomenys, virtualios esybės funkcionalumas, gebėjimas sąveikauti ir esminiai duomenys, duomenų mainus nusakanti kryptis ir duomenų sinchronizacija.

Padėka

Autorius dėkoja Duomenų mokslo ir skaitmeninių technologijų instituto doc. A. Lupeikienei už patarimus atliekant tyrimus ir rengiant šį straipsnį.

Literatūra

[1] Liu, M., Fang, S., Dong, H., Xu, C. (2021) Review of digital twin about concepts, technologies, and industrial applications. Journal of Manufacturing Systems, Volume 58, Part B, 346-361.

[2] Durão, L.F.C.S., Haag, S., Anderl, R., Schützer, K., Zancul, E. (2018). Digital Twin Requirements in the Context of Industry 4.0. In: Chiabert, P., Bouras, A., Noël, F., Ríos, J. (eds) Product Lifecycle Management to Support Industry 4.0. PLM 2018. IFIP Advances in Information and Communication Technology, vol 540. Springer, Cham.

[3] Van der Valk, H.; Henning, J.-L.; Winkelmann, S.; Haße, H. (2022) Practical Requirements for Digital Twins in Production and Logistics. In: Herberger, D.; Hübner, M. (Eds.): Proceedings of the Conference on Production Systems and Logistics: CPSL 2022. Hannover: publish-Ing., 32-41.

[4] B. R. Barricelli, E. Casiraghi and D. Fogli (2019) A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications," in *IEEE Access*, vol. 7, 167653-167671.

[5] Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P., ... Moher, D. (2009). The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *Annals of internal medicine*, *151*(4), W-65.

[6] Glaessgen, E., Stargel, D. (2012). The digital twin paradigm for future NASA and US Air Force vehicles. In 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, p. 1818.

[7] Minerva, R., Lee, G. M., Crespi, N. (2020). Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models. Proceedings of the IEEE, 108(10), 1785-1824.

[8] Rajagopal H. (2023) Digital Twin: Virtual Model of Real-World Entity. Randamas adresu: https://www.linkedin.com/pulse/digital-twin-hariprasanth-rajagopal.

[9] Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. *White paper*, *1*(2014), 1-7.

[10] Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology, 94, 3563-3576.

[11] Liu, Z., Meyendorf, N., Mrad, N. (2018). The role of data fusion in predictive maintenance using digital twin. In AIP conference proceedings, Vol. 1949, No. 1. AIP Publishing.
[12] Zhang, H., Liu, Q., Chen, X., Zhang, D., Leng, J. (2017). A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. IEEE Access, 5, 26901-26911. [13]
[13] Weber, C., Königsberger, J., Kassner, L., Mitschang, B. (2017). M2DDM–a maturity model for data-driven manufacturing. Procedia Cirp, 63, 173-178.

Elektromobilių baterijų likutinės vertės prognozavimas

Kasparas Veličkevičius

Vilniaus universitetas, Taikomosios matematikos institutas,
Naugarduko g. 24, Vilnius
kasparas.velickevicius@gmail.com

Santrauka. Elektromobilio baterijos likutinė vertė (angl. *state of health*, *SoH*) yra baterijos tikrosios elektrinės talpos ir gamintojo nustatytos talpos santykis. Šis rodiklis apibūdina baterijos degradaciją. Nuo tikslaus baterijos *SoH* įvertinimo priklauso automobilio efektyvumas ir saugumas. Šio darbo tikslas – prognozuoti baterijos likutinę vertę, naudojant krovimo įtampos duomenis. Tam parinktas *XGBoost* modelis, gebantis prognozuoti elektromobilių baterijų elementų *SoH*, remiantis krovimo įtampos kreivių požymiais.

Raktiniai žodžiai: elektromobiliai, baterijos, įtampos kreivės, *SoH*, *XGBoost*

1 Įvadas

Pastaraisiais dešimtmečiais pasaulyje kyla vis rimtesnių problemų, susijusių su globaliniu atšilimu ir iškastinio kuro išteklių trūkumu, todėl žmonės vis labiau domisi švarios energijos naudojimu. Tai suteikia poreikį plėtoti elektromobilius, kurie plačiai pripažįstami kaip svarbi priemonė kovojant su aplinkos tarša ir energijos resursų stoka. Ličio jonų baterijos plačiai naudojamos kaip elektromobilių energijos šaltinis dėl jų efektyvumo. Tačiau jų veikimas laikui bėgant blogėja dėl senėjimo ir kitų veiksnių, o tai sumažina baterijų energijos kaupimo ir energijos perdavimo pajėgumą. Tikslus baterijos būklės (*SoH*) įvertinimas tapo svarbiu uždaviniu elektromobilių pramonėje, nes tai lemia saugų ir efektyvų transporto priemonės funkcionavimą.

Šiame darbe nuspręsta prognozuoti *SoH*, nagrinėjant kelių įtampos kreivės regionų (krovimo ir atsipalaidavimo) požymius kartu, o ne atskirai, kaip įprasta literatūroje. Taip siekiama išnaudoti visą krovimo įtampos kreives informaciją ir potencialiai gauti didesnį prognozavimo tikslumą.

2 Literatūros apžvalga

SoH vertinimas plačiai sprendžiamas uždavinys literatūroje. [2, 3] straipsniuose tirti baterijos elektros stiprio, įtampos, temperatūros kreivių požy-

DOI: https://doi.org/10.15388/LMITT.2024.19

miai. *SoH* prognozavimui autoriai naudojo *RVM*, neuroninius tinklus. [4, 5] straipsniuose pagrįsta, kad praktiškiausia tirti krovimo duomenis, nes jie mažiau priklauso nuo automobilio vairavimo ypatumų, lyginant su iškrovos duomenimis. Kaip modelio požymius autoriai naudojo laiką įvairiuose voltų intervaluose, pastovaus stiprio krovimo metu. Parinkti *RFR*, *LS-SVM* modeliai. [6, 7] straipsniuose iš įtampos kreivių apskaičiavo tokius požymius, kaip pastovaus stiprio krovimo laikas, pastovios įtampos krovimo laikas, įtampos kreivės krypties koeficientas fiksuotam voltų intervale. Prognozavimui parinkti *GPR* ir *LSTM* modeliai. [8] straipsnyje pagrįsta, kad atsipalaidavimo periodas po įkrovimo stipriai susijęs su *SoH*. Anot autorių, naudinga tirti šį įtampos kreivės regioną, nes jis mažiau priklauso nuo krovimo sąlygų. Iš įtampos kreivių atsipalaidavimo metu apskaičiuoti statistiniai požymiai, *SoH* prognozavimui parinkti *SVR* ir *XGBoost* modeliai.

3 Duomenys

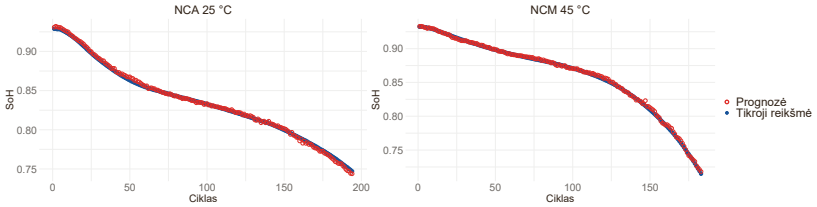
Duomenys paimti iš viešai prieinamo duomenų rinkinio, kuris buvo naudojamas [8] straipsnyje. Duomenys surinkti laboratorijoje testuojant komerciškai prieinamas ličio jonų baterijas. Testuotos 105 dviejų cheminių sudėčių baterijos: nikelio-kobalto-aliuminio (*NCA*) ir nikelio-kobalto-mangano (*NCM*). Testai atlikti kamerose su kontroliuojama temperatūra (25, 35 ir 45 °C). Naudotas pastovaus stiprio – pastovios įtampos (*CCCV*) krovimo protokolas ir pastovaus stiprio iškrova. Vienas baterijos testavimo ciklas apima penkis procesus: pastovaus stiprio krovimą, pastovios įtampos (*CV*) krovimą, 30 min atsipalaidavimą po įkrovimo, *CC* iškrovimą ir 30 min atsipalaidavimą po iškrovimo.

Tirti šie požymiai: *F*1 – pastovaus stiprio krovimo laikas, *F*2 – pastovios įtampos krovimo laikas, *F*3 – įtampos kreivės krypties koeficientas voltų intervale [4; 4,2] pastovaus stiprio krovimo metu; *F*4 – maksimali įtampos reikšmė atsipalaidavimo periode; *F*5 – minimali įtampos reikšmė atsipalaidavimo periode; *Ch* – baterijos cheminė sudėtis; *T* – baterijos testavimo aplinkos temperatūra.

4 Rezultatai ir išvados

Tikslui įgyvendinti parinktas *XGBoost* modelis. Tai ansamblinio mokymo algoritmas, paremtas gradientų tobulinimo pasirinkimų medžiais. Šis metodas parenka silpnus sprendimų medžius paeiliui ir kiekvienas medis pako-

reguoja prieš tai parinkto medžio prognozes [1]. Buvo optimizuoti šie modelio hiperparametrai: maksimalus medžio gylis, minimalus mazgo svoris, dalinės imties dydis, požymių dalis medyje, žingsnio dydis, medžių skaičius. Hiperparametrų optimizavimui naudotas *Grid search* algoritmas. Modelio prognozavimo tikslumo testinėje aibėje metrikos *MAE* = 0,0022, *MAPE* = 0,0027, *RMSE* = 0,0030. 1 pav. pateiktas grafikas, rodantis mažas paklaidas tarp prognozuotų ir tikrųjų reikšmių. Taigi šiame darbe parinktas modelis, gebantis aukštu tikslumu prognozuoti elektromobilio baterijos likutinę vertę, remiantis baterijos krovimo įtampos duomenimis.



1 pav. Dviejų testinės aibės baterijų tikrosios ir prognozuotos SoH reikšmės, didėjant testavimo ciklų skaičiui.

Literatūra

[1] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. 2016.

[2] C. Du, R. Qi, Z. Ren, and D. Xiao. Research on state-of-health estimation for lithiumion batteries based on the charging phase. Energies, 16(3):1420, Feb. 2023.

[3] P. Guo, Z. Cheng, and L. Yang. A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction. J. Power Sources, 412:442–450, Feb. 2019.

[4] C. Lin, J. Xu, M. Shi, and X. Mei. Constant current charging time based fast state-of-health estimation for lithium-ion batteries. Energy (Oxf.), 247(123556):123556, May2022.

[5] X. Shu, G. Li, J. Shen, Z. Lei, Z. Chen, and Y. Liu. A uniform estimation framework for state of health of lithium-ion batteries considering feature extraction and parameters optimization. Energy (Oxf.), 204(117957):117957, Aug. 2020.

[6] D. Yang, X. Zhang, R. Pan, Y. Wang, and Z. Chen. A novel gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. J. Power Sources, 384:387–395, Apr. 2018.

[7] U. Yayan, A. T. Arslan, and H. Yucel. A novel method for SoH prediction of batteries based on stacked LSTM with quick charge data. Appl. Artif. Intell., 35(6):421–439, May 2021.

[8] J. Zhu, Y. Wang, Y. Huang, R. Bhushan Gopaluni, Y. Cao, M. Heere, M. J. Mühlbauer, L. Mereacre, H. Dai, X. Liu, A. Senyshyn, X. Wei, M. Knapp, and H. Ehrenberg. Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. Nat. Commun., 13(1):2261, Apr. 2022.

Unit Test Generation Using Large Language Models: A Systematic Literature Review

Dovydas Marius Zapkus, Asta Slotkienė

Vilnius University, Universiteto g. 3, Vilnius
marius.zapkus@mif.stud.vu.lt, asta.slotkiene@mif.vu.lt

Abstract. Unit testing is a fundamental aspect of software development, ensuring the correctness and robustness of code implementations. Traditionally, unit tests are manually crafted by developers based on their understanding of the code and its requirements. However, this process can be time-consuming, error-prone, and may overlook certain edge cases. In recent years, there has been growing interest in leveraging large language models (LLMs) for automating the generation of unit tests. LLMs, such as GPT (Generative Pre-trained Transformer), CodeT5, StarCoder, LLaMA, have demonstrated remarkable capabilities in natural language understanding and code generation tasks. By using LLMs, researchers aim to develop techniques that automatically generate unit tests from code snippets or specifications, thus optimizing the software testing process. This paper presents a literature review of articles that use LLMs for unit test generation tasks. It also discusses the history of the most commonly used large language models and their parameters, including the first time they have been used for code generation tasks. The result of this study presents the large language models for code and unit test generation tasks and their increasing popularity in code generation domain, indicating a great promise for the future of unit test generation using LLMs.

Keywords: unit test generation, large language model

1 Introduction

Software testing is one of the most important software development processes, which increases the overall quality and reliability of the final product [1].

Unit testing is an essential part of software testing methods. Successful implementation of unit tests can decrease the number of errors in the final product and increase the efficiency of the developer, thus making

DOI: https://doi.org/10.15388/LMITT.2024.20

the software more reliable [2, 9]. Unit tests are used to test units of code, ranging from functions, methods, procedures, etc. [3]

To optimize the testing of information systems, automated testing tools are applied, which help developers save resources and time [2]. Currently, solutions such as Search-Based Software Testing [4] and random test generation [5, 14] are used for software testing. These solutions are capable of generating component tests, but often the generated tests are impractical and difficult to understand [4-6]. Therefore, it has been suggested to use large language models, which would not only effectively cover system functionality but also be clear and easily interpretable. Currently, programs utilizing large language models, such as Github Copilot, can successfully generate code from code comments or complete the remaining part of a started code segment [7, 12]. Consequently, it is believed that large language models can also be employed in unit or component testing.

Upon analyzing scientific research [14, 16] on large language models and their generated component tests, it is observed that models such as OpenAI GPT-3 and Codex are the most commonly used LLMs for unit test code generation tasks. These models are already being used in other domains, and their potential and effectiveness in the context of test case generation have not been thoroughly explored.

This paper aims to perform a literature review on studies that analyze LLMs for unit test generation tasks, to identify which large language models are used for these tasks and what are the new relationships between large language models and unit test domains since 2019.

The rest of this paper is structured as follows. Section 2 presents the review methodology. Section 3 discusses the results obtained in the paper. Finally, Section 4 concludes the paper.

2 Research methodology

The review methodology was developed and executed according to the guidelines provided by Kitchenham and Brereton [17, 18]. The structure of the methodology consists of five steps: (1) preparing of review, (2) identification, (3) screening, (4) eligibitation (5) developing mapping and analysis (Figure 1).

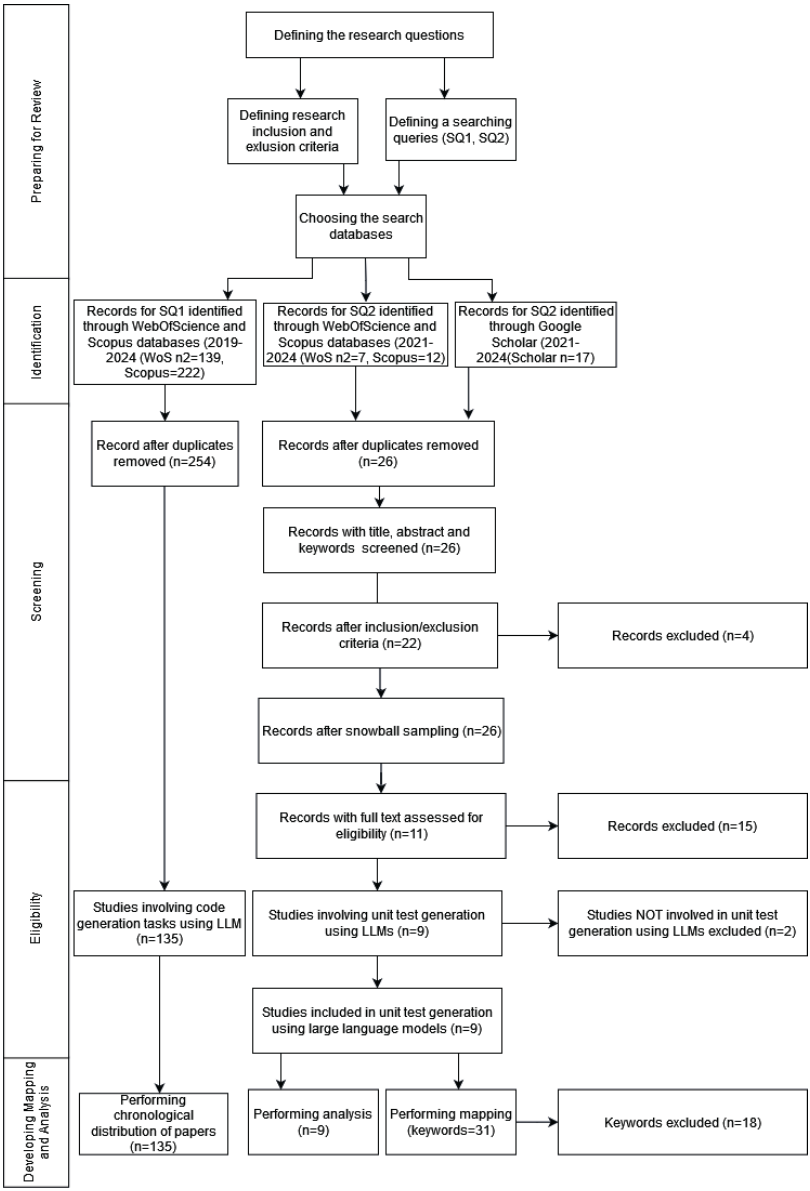


Figure 1. The flow diagram of the systematic literature review

In the first step, we raised two research questions, which covered the main research question: *What is covered for unit test generation using large language models?*

RQ1: What is the historical evolution of code (unit test) generation using large language models?

RQ1.1 What kind of large language models are used for code generation?

RQ1.2 Which large language models are used for unit test generation tasks?

RQ2: What are the new relationships between large language models and unit test domains in the analyzed period?

To increase the research accuracy we determined the inclusion and exclusion criteria (Table 1):

Studies Inclusion Criteria (IC)	Studies Exclusion Criteria (EC)
IC1. Studies were published after 2021 **IC2.** The publication must be written in English **IC3.** The publication is a primary study **IC4.** Studies that compare large language models for unit test generation tasks. **IC5.** Studies are in the field of software engineering or computer science **IC6.** Studies which have an empirical background	**EC1**. Studies that are duplicates of other studies of the same authors **EC2**. The reported research does not relate to LLM and unit tests, or the research is not discussed in the context of LLM unit test generation **EC3**. The publication was not written in English **EC4**. Studies not accessible in full-text

Table 1. Inclusion/Exclusion criteria

The search strategy includes two main terms, which help to define the search queries: The final search query is developed based on WoS, Scopus, and Google Scholar databases search requirements as follows:

1. The SQ1 was created and applied as follows: specify the primary search keywords based on the main research question; find substitute alternatives for the large language models (LLMAlternatives) such as GTP-4, Codex, etc.

   SQ1: *LLMAlternatives AND code AND (generation OR automation)*

2. Search query (SQ2) decides RQ1.2 and RQ2 and covers terms related to unit tests: unit test and component test and terms related to large language models: large language model and LLM.

   SQ2: *("unit test*" OR "component test*") AND ("large language model*" OR LLM OR LLM's)*

The identification step was performed the search using the defined search query in three scientific databases such as WoS and Scopus, and Google Scholar (databases set of scientific papers), using these limitations: articles written in only English language (IC2), the document type was only articles, the article should be covered only subject area Computer Science. After searching by selected search query were obtained 37 articles from 2019 to 2023. Because the same search query was performed in three different databases, in the s*creening step, was removed* duplicated studies (11 studies removed) and records were verified by inclusion/exclusion criteria (4 studies excluded). A deeper analysis of the found articles allowed us to notice that some articles we knew about weren't included in the search results. That way, we used snowball sampling and added several scientific articles (4 studies), which included the mentioned terms and the main scientific question. An interim analysis of the keyword map from 9 articles indicated the need to apply a relevancy analysis of full-text articles (eligibility step). The full text of each study is assessed for eligibility (15 excluded) and each study is validated based on whether it performs research on unit test generation using large language models (2 excluded).

3 Results of systematic literature review

In the emerging application of using Large Language Models for generating unit tests and the limited research in this domain, this study instead aims to examine the increasing adoption of LLMs in code generation tasks. By performing a literature review in Section 2 it was observed, that only 9 studies were retrieved. This result wouldn't yield sufficient data to analyze the popularity of LLM's. A more abstract term „code generation" was selected to retrieve a larger number of studies (SQ1), improving the overall accuracy of the chronological distribution of the papers diagram in Figure 2. „Unit test generation" is a subtopic of „code generation tasks" [7], thus we can assume, that LLM's ability to generate code makes it able to generate unit tests. The chronological distribution of papers on LLM for code generation tasks is shown in Figure 2. The papers selected were published from 2019 till the end of 2024 (RQ1).

From Figure 2 we can notice, that the highest number of studies related to LLM and code generation tasks is observed in the year 2023 with a total of 135 studies, contributing to 54.87% of the overall number of studies since the year 2019 (RQ1.1). This indicates that research on LLMs and code

generation tasks has been growing in popularity over the years with an average of 56.82% annually. The number of publications for each LLM in Figure 2 indicates, that these models can be used for code generation tasks (RQ1.1). Codex, released in 2021, is the highest-researched LLM for code generation tasks included in 21.9% of all the studies analyzed in Figure 2. Another popular LLM is GPT-2 having an equal number of research papers as Codex, but was released in 2019 and indicates a stagnation or decrease in the number of papers since 2022. Another highly researched LLM is GPT-3, having 50 studies since 2020 and gaining popularity annually. The highest number of LLMs with the ability to generate code was released in 2023.

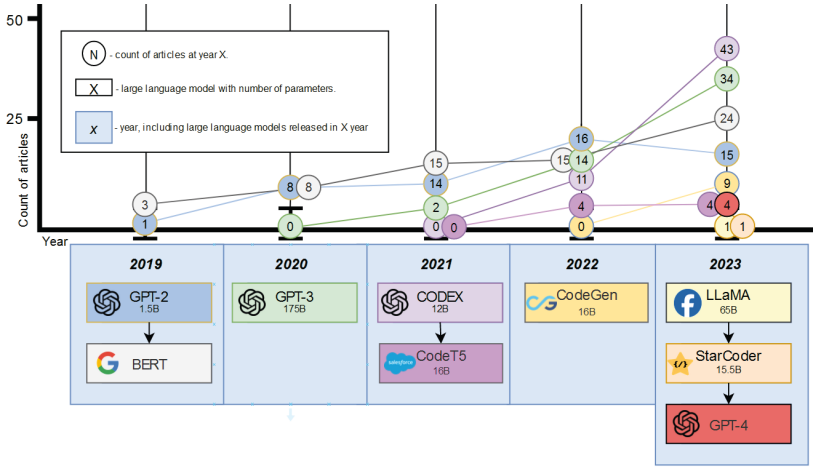


Figure 2. Chronological distribution of papers by large language models used for code generation tasks.

In addition to RQ1.2. From gathered papers [8-16] using the literature review in Figure 1, it was noticed that for unit test generation tasks, models such as Codex, GPT-3, StarCoder, and GPT-4 were used. Figure 2 indicates the rising popularity of LLM's ability to generate code. These results indicate that with the rising popularity of LLM, more distinct topics are selected for research of this model, such as unit test generation, while less researched models are excluded from such topics.

For the second research question (RQ2), it was decided to perform a keyword map from articles gathered in Figure 1. The creation of a keyword

map was selected for its ability to distinguish associations between different keywords and group these keywords into clusters, thus improving the analysis process. The keyword map was developed by using the VOSviewer tool. The original result contained 49 unique keywords. It was noticed that some of the keywords didn't indicate the contextual relationship between LLM and unit tests, these keywords include research type („review", „literature survey", „literature review"), abstract keywords („agenda", „group", „focus", „codes"). After removing these keywords, 31 items remained and a keyword map was developed using VOSviewer in Figure 3.

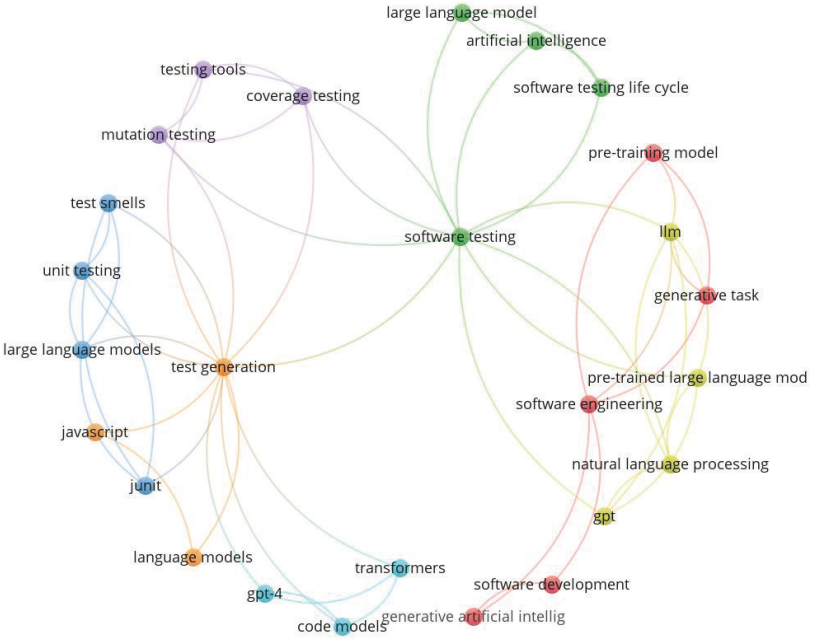


Figure 3. LLM and unit test keywords map

Generated keyword map in Figure 3, aims to answer RQ2. We can see that the 6 clusters were formed. The most frequent keywords were "test generation" and „software testing". Most associations to other clusters were made through "software testing" keyword. "Test generation" keyword is connected with „gpt-4", „large language models", „code models" keywords, which indicates the type of tools/models that were used for test generation

tasks in the studies. “Test generation” keyword also associated with type of unit test quality testing, which is in a color purple cluster. „Software testing“ is associated with “llm” keyword that indicates, that studies were mostly using large language models for testing software.

Summing up, according to the historical overview, the analyzed topic of the unit test generation using large language models is quite new based on the scarcity of articles surrounding it, but it is becoming more and more frequent and since 2023, it has had the highest increase in its relevance. Most studies in recent years have performed research with Codex and GPT-3 large language models.

4 Conclusion and Future Work

This study is a systematic literature review of studies researching large language models and their capability to generate unit tests. It was noticed that LLM and unit test generation topics are new, and related research on these topics is relatively tiny. LLM and code generation tasks have been gaining popularity since 2019, with an average increase of 56.82% annually. From deeper analysis, we can indicate that models such as Codex, GPT-3, StarCoder, and GPT-4 were used for unit test generation tasks. From generating the keywords map, it was noticed that unit test generation has relationships not only with various AI artifacts but also relevant to the quality aspects of the test generation process and test quality.

The results of this study indicate future works in the analyzed area. First, the quality criteria are determined, which helps to evaluate the quality of generated unit tests. The second challenge is research, which shows the activity chain for the test development process using LLMs.

Literature

[1] Nagabushanam, Durga & Dharinya, Sree & Vijayasree, Dasari & Sai Roopa, Nadendla & Arun, Anugu. (2022). A Review on the Process of Automated Software Testing. 10.48550/arXiv.2209.03069.

[2] Job, Minimol. (2021). Automating and Optimizing Software Testing using Artificial Intelligence Techniques. International Journal of Advanced Computer Science and Applications. 12. 10.14569/IJACSA.2021.0120571.

[3] Nikolaeva Zheleva, Dimitrichka. (2021). The role of unit testing in training. In: Development Through Research and Innovation - 2021 [online]: The 2nd International Scientific Conference: Online Conference for Researchers, PhD and Post-Doctoral Students, Au-

gust 27th, 2021, Chişinău: Conference Proceedings. Chişinău, ASEM, 2021, pp. 42-49. ISBN 978-9975-155-54-0.
[4] Fontes, A., & Gay, G. (2023). The integration of machine learning into automated test generation: A systematic mapping study. Software Testing, Verification and Reliability, 33(4), e1845, 10.48550/arXiv.2206.10210.
[5] Hashtroudi, Sepehr & Shin, Jiho & Hemmati, Hadi. (2023). Automated Test Case Generation Using Code Models and Domain Adaptation.
[6] Grano, Giovanni & Palomba, Fabio & Nucci, Dario & Lucia, Andrea & Gall, Harald. (2019). Scented Since the Beginning: On the Diffuseness of Test Smells in Automatically Generated Test Code. Journal of Systems and Software. 156. 10.1016/j.jss.2019.07.016.
[7] Bareiß, Patrick & Souza, Beatriz & d'Amorim, Marcelo & Pradel, Michael. (2022). Code Generation Tools (Almost) for Free? A Study of Few-Shot, Pre-Trained Language Models on Code. 10.48550/arXiv.2206.01335.
[8] Siddiq, M. L., Santos, J., Hasan Tanvir, R., Ulfat, N., Al Rifat, F., & Carvalho Lopes, V. (2023). Exploring the effectiveness of large language models in generating unit tests. arXiv e-prints, arXiv-2305.
[9] Alagarsamy, S., Tantithamthavorn, C., & Aleti, A. (2023). A3test: Assertion-augmented automated test case generation. arXiv preprint arXiv:2302.10352.
[10] Le, H., Wang, Y., Gotmare, A. D., Savarese, S., & Hoi, S. C. H. (2022). Coderl: Mastering code generation through pre-trained models and deep reinforcement learning. Advances in Neural Information Processing Systems, 35, 21314-21328.
[11] Bayrı, V., & Demirel, E. (2023, December). AI-Powered Software Testing: The Impact of Large Language Models on Testing Methodologies. In 2023 4th International Informatics and Software Engineering Conference (IISEC) (pp. 1-4). IEEE.
[12] Huang, Y., Chen, Y., Chen, X., Chen, J., Peng, R., Tang, Z., Huang, J., Xu F. & Zheng, Z. (2024). Generative Software Engineering. arXiv preprint arXiv:2403.02583.
[13] Nguyen-Duc, A., Cabrero-Daniel, B., Przybylek, A., Arora, C., Khanna, D., Herda, T., ... & Abrahamsson, P. (2023). Generative Artificial Intelligence for Software Engineering--A Research Agenda. arXiv preprint arXiv:2310.18648.
[14] Wang, J., Huang, Y., Chen, C., Liu, Z., Wang, S., & Wang, Q. (2024). Software testing with large language models: Survey, landscape, and vision. IEEE Transactions on Software Engineering.
[15] Guilherme, V., & Vincenzi, A. (2023, September). An initial investigation of ChatGPT unit test generation capability. In Proceedings of the 8th Brazilian Symposium on Systematic and Automated Software Testing (pp. 15-24).
[16] Schäfer, M., Nadi, S., Eghbali, A., & Tip, F. (2023). An empirical evaluation of using large language models for automated unit test generation. IEEE Transactions on Software Engineering.
[17] Kitchenham, Barbara & Brereton, Pearl & Budgen, David & Turner, Mark & Bailey, John & Linkman, Stephen. (2009). Systematic literature reviews in software engineering-A systematic literature review. Information and Software Technology. 51. 7-15. 10.1016/j.infsof.2008.09.009.
[18] Kitchenham, B. and Brereton, P. (2013) A Systematic Review of Systematic Review Process Research in Software Engineering. Information and Software Technology, 55, 2049-2075.

Klaidingų iškvietimų identifikavimas

Eimantas Zaranka, Rūta Juozaitienė, Tomas Krilavičius

Vytauto Didžiojo Universitetas,
Universiteto g. 10, 53361 Akademija, Kauno r.
Center of Applied Research and Development (CARD),
Universiteto g. 10, 53361 Akademija, Kauno r.
eimantas.zaranka@vdu.lt

Santrauka. Reagavimas į klaidingus iškvietimus trikdo ne tik sklandų saugos paslaugų veikimą, bet ir eikvoja energijos išteklius, didina išmetamų ŠESD emisiją bei transporto atliekų susidarymą. Šio tyrimo tikslas sukurti klaidingų iškvietimų aptikimo modelį, kuris leistų efektyviau valdyti reagavimo į iškvietimus procesą. Pasiūlyta modeliavimo strategija remiasi ansamblinio mašininio mokymosi metodais bei pasižymi gana aukštu tikslumu, F_1 įverčio reikšmė lygi 0,887. Imituojant realias aplinkos sąlygas buvo atliktas eksperimentinis pasiūlytos metodikos testavimas, tokiu būdu patvirtinant jos efektyvumą bei prognozavimo tikslumą.

Raktiniai žodžiai: XGBoost, ansamblinis modelis, klasifikavimas, klaidingi iškvietimai, objektų sauga

1 Įvadas

Klaidingas iškvietimas nusako situaciją, kai atliekamas iškvietimas į manomai pavojingą ar nukrypstančią nuo standartų situaciją, tačiau išsiuntus reagavimo ekipažą pastebima, kad poreikis į reagavimą nėra reikalingas. Su klaidingais iškvietimais dažniausiai susiduriama apsaugos ir saugumo sistemose, kuriose naudojami judesio, įsilaužimo davikliai ir stebėjimo kameros, kurios reaguodamos į įvairius aplinkos veiksnius perduoda iškvietimo signalą į operacijų centrą. Tačiau galimi sistemos gedimai, gyvūnų elgesys ar kiti veiksniai gali sąlygoti klaidingus iškvietimus, kurie trikdo sklandų saugos paslaugų veikimą, turi neigiamą įtaką efektyviam resursų naudojimui bei mažina pasitikėjimą saugos sistemomis.

Siekiant gerinti siūlomų saugos paslaugų kokybę, atsiranda poreikis diegti klaidingų iškvietimų analizės sistemas. Tokio pobūdžio sistemos yra naudingos saugos tarnyboms, viešosios tvarkos tarnyboms, specialiosioms bei panašaus pobūdžio tarnyboms kadangi padeda efektyviai valdyti reagavimo į iškvietimus procesą, bei ženkliai sumažinti transporto naudojimą –

DOI: https://doi.org/10.15388/LMITT.2024.21

taupyti energijos išteklius (kurą, elektros energiją), mažinti išmetamų ŠESD emisiją, mažinti transporto atliekų susidarymą.

Šiame straipsnyje pristatomi eksperimentiniai tyrimai, kurių tikslas remiantis iškvietimų į objektus istoriniais duomenimis, sudaryti tikimybinį klaidingų iškvietimų aptikimo modelį. Atlikta analizė apima problemos analizę, duomenų paruošimą ir pradinę analizę, svarbių požymių identifikavimą bei eksperimentus su skirtingais klasifikavimo modeliais. Remiantis eksperimentų metu gautais rezultatais, sukurtas klaidingų iškvietimų aptikimo modulio maketas.

2 Literatūros apžvalga

Straipsnyje [1] pristatomas hibridinis iškvietimų (angl. *alarm*) vertinimas. Autorių teigimu, siūlomas sprendimas sugeba įvertinti 30 tūkst. iškvietimų per sekundę, taip sprendžiant sistemos perkrovos ir patikimumo gerinimo užduotis. Tyrime [1] naudojami *Sitasys*, *London* ir *San Francisko* tikrų iškvietimų duomenų rinkiniai. Dirbtinio intelekto modeliui kurti naudoti keturi algoritmai: atsitiktiniai medžiai (angl. r*andom forest*), logistinė regresija (angl. *logistic regression*), atraminių vektorių mašina (angl. *support vector machine*) ir neuroninis tinklas. Atlikus eksperimentus pastebėta, kad geriausiai veikia atsitiktinių medžių klasifikatorius, kuriuo pasiekiamas 92 % tikslumas. Tyrime pabrėžiama, kad mašininio mokymosi modeliai turėtų būti kuo paprastesni, siekiant efektyviai dirbti su didelio srauto duomenimis.

Straipsnyje [2] pristatomas įsilaužimų į pastatus aptikimas, naudojant *Wi-Fi* kanalų būsenos (angl. *channel state*) informaciją. Autoriai atlieka klasifikavimą naudojant atraminių vektorių klasifikatorių, kurio pagalba nustatoma ar buvo įsilaužta pro langą ar pro duris. Sudarytas modelis buvo ištestuotas atliekant daugiau nei 200 eksperimentų, kurių metu nustatyta, kad jeigu namie nėra pašalinio judesio, sukurtas modelis veikia 94,5 % tikslumu, jeigu juda vienas pašalinis asmuo sistema veikia 84 %, kur 1,7 % yra klaidingi iškvietimai, atvejais kai užfiksuojamas dviejų asmenų judėjimas modelis veikia 69 % tikslumu.

Straipsnyje [3] pristatomas klaidingų skubios pagalbos iškvietimų nustatymas, analizuojant skambinančiojo elgseną ir vietą. Tyrime pristatomas ribine verte paremtas vertinimas, siekiant išskirti klaidingus iškvietimus. Kiekvienam objektui priskiriamas patikimumo koeficientas *f*, kuris atnaujinamas, po kiekvieno atlikto iškvietimo pagal tai ar tas iškvietimas pasitvirtino, ar ne. Inicializuojama *f* reikšmė lygi 0. Atveju, kada iškvietimas yra klaidingas

f reikšmė padidinama 1. Šis koeficientas yra lyginamas su dviem slenkstinėmis ribomis f_1 ir f_2, kurios naudojamos skambinančiojo patikimumo įvertinimui. Patikimi skambinantieji laikomi tie, kuriems $f \leq f_1$, ir įtartini, kurie $f_1 \leq f \leq f_2$, o ignoruojami, tie kuriems $f \geq f_2$. Sistema veikia pasitikėjimo principu, todėl patikimi skambinantieji nėra analizuojami.

Straipsnyje [4] pristatomas klaidingų iškvietimų aptikimas naudojant statistinius metodus. Tyrime, taikomiems algoritmams įvertinti naudojama klaidingai teigiamų rezultatų dažnis (angl. *false positive rate*), tikslumas (angl. *precision*), teisingumas (angl. *accuracy*), jautrumas (angl. *sensitivity*) ir specifiškumas (angl. *specificity*). Modelių kūrimui naudojama trijų etapų strategija: sensorių istorinių duomenų, kurie veikia normaliomis sąlygomis, surinkimas ir statistinio modelio sudarymas, kontrolinės ribos nustatymas sudarytam modeliui ir gyvo srauto stebėjimas. Straipsnyje pastebima, kad nors ir egzistuoja nemažai tyrimų analizuojančių klaidingų iškvietimų aptikimą, tačiau optimalus sprendimas, kuris leistų praktikoje identifikuoti tokius iškvietimus, nerastas.

Straipsnyje [5] pristatomas iškvietimų požymių aptikimas, dideliame iškvietimų duomenų sraute. Anot autorių dideli srautai yra viena pagrindinių priežasčių, kodėl iškvietimų sistemos pasižymi prastu efektyvumu ir duoda mažai naudos. Autoriai naudoja uždarų asociatyvinių taisyklių išskyrimą (angl. *closed association rule mining*, *CHARM*). Siūlomas sprendimas veikia trimis etapais: didelio srauto nustatymas, uždarų šablonų (angl. *patterns*) nustatymas ir atitinkamų šablonų nustatymas. Vidutinis algoritmo veikimo laikas 2 minutės. Autoriai teigia, kad siūlomas sprendimas yra veiksmingas ieškant pavojaus signalų šablonų bei padeda sumažinti srauto perteklių.

3 Tyrimo metodai

Tyrime naudoti logistinės regresijos, Gauso maišos, SGD, artimiausių kaimynų, sprendimų ir atsitiktinių medžių miško, gradientinio didinimo, LGBM, XGBoost klasifikatoriai ir ansamblinis modelių junginys.

XGB metodas yra vienas iš dažniausiai naudojamų gradiento didinimo sprendimų medžių algoritmų. Jo atveju prognozės tikslumas patobulinamas kuriant daugybę modelių ir akcentuojant tuos mokymo atvejus, kurie yra sunkiai įvertinami. Taip generuojami pradiniai modeliai skirstomi į dvi grupes: silpnus ir stiprius. Silpnas modelis yra algoritmas, kuris veikia tik šiek tiek geriau nei atsitiktinis spėjimas, tuo tarpu stiprus modelis yra tikslesnis prognozavimo ar klasifikavimo algoritmas, kuris yra stipriai koreliuotas su

sprendžiama problema. *XGB* modeliuose silpnaisiais modeliais laikomi pradiniai sprendimų medžiai. Kiekvienas medis bando sumažinti ankstesniojo klaidas. Nors augantys medžiai yra silpnai besimokantys, tačiau pridedant daug medžių iš eilės ir kiekviename iš jų atsižvelgiant į praeitų medžių klaidas, padidėja efektyvumas ir modelio tikslumas. Kadangi medžiai yra pridedami nuosekliai, mokymosi algoritmas yra lėtas, tačiau tikslus.

Sudarytų klientų atmetimo prognozavimo modelių tikslumui įvertinti buvo pasirinkti plačiai taikomi tikslumo matai: tikslumas (angl. *precision*), išsamumas (angl. *recall*) ir F_1 kriterijus. Šie matai apskaičiuojami naudojant maišaties matricą (angl. *confusion matrix*), kuri apibendrina klasifikavimo algoritmų efektyvumą ir yra sudaryta iš teisingų teigiamų (angl. *true positive*, *TP*), teisingų neigiamų (angl. *true negative*, *TN*), klaidingų teigiamų (angl. *false positive*, *FP*) ir klaidinagų neigiamų (angl. *false negative*, *FN*) atvejų skaičių. Tikslumas gali būti apibrėžtas kaip modelio klasifikavimo kokybės matas apskaičiuojamas remiantis formule:

$$precision = \frac{TP}{TP+FP}.$$

Išsamumas apibrėžiamas kaip santykis tarp teisingai suklasifikuotų teigiamų atvejų skaičiaus ir visų teigiamų atvejų skaičiaus:

$$recall = \frac{TP}{FN+TP}.$$

F_1 matas yra tikslumo ir išsamumo matų harmoninis vidurkis. Apskaičiuojant F_1 įvertį yra įtraukiamos neteisingai suklasifikuotos reikšmės, todėl šis matas nusako, kiek tikslus ir griežtas yra klasifikatorius atsižvelgdamas ne tik į teisingai suklasifikuotus atvejus, bet ir į blogai suklasifikuotus. F_1 matas gali būti apskaičiuojamas remiantis formule:

$$F1 = 2 * \frac{precision*recall}{precision+recall}.$$

4 Tyrimo duomenys

Šiame tyrime analizuojami istoriniai saugos tarnybos iškvietimų duomenys. Duomenų aibę sudaro informacija apie 12 598 objektus stebimus 2019 gruodžio 1 d. – 2021 lapkričio 30 d. laikotarpiu. Analizuojami duomenys apima: informaciją apie iškvietimus ir stebimus objektus, meteorologinę informaciją bei nusikalstamas veiklas analizuojamoje lokacijoje.

Pirmojoje kategorijoje fiksuojama informacija apie atliktų iškvietimų laiko žymes (angl. *timestamp*) ir reagavimą į gautą iškvietimą, reaguojamo

objekto miestą, adresą, rajoną, geografines koordinates ir objekto paskirtį. Antrojoje kategorijoje pateikiama oro temperatūra, matuojama Celsijais, vėjo greitis, metrais per sekundė ir kritulių kiekis milimetrais. Trečiojoje kategorijoje pateikiama informacija apie nusikalstamų veikų pasiskirstymą pagal lokacijas. Šie duomenys gaunami iš nusikalstamų veikų žinybinio registro duomenų žemėlapio[1]. Šiame tyrime naudojami duomenys apima nusikaltimo datą, nusikalstamos veikos tipą ir užfiksuotos veikos lokaciją (koordinates), pagal kurią apskaičiuojamas nusikaltimų skaičius analizuojamo objekto aplinkoje.

5 Pradinė duomenų analizė

Siekiant susipažinti su turimu duomenų rinkiniu, įgauti įžvalgų apie duomenų pasiskirstymą bei sąryšius tarp požymių, buvo atlikta pradinė duomenų analizė. Duomenys gaunami iš trijų srautų, todėl pradinės duomenų analizė pradėta nuo pateiktų duomenų apjungimo. Apjungtą duomenų rinkinį sudaro 289 337 stebiniai (iškvietimai), užfiksuoti 12 598 objektams. Kiekvieną stebėjimą apibūdina 11 požymių, kurie priklauso keturioms anksčiau pristatytoms kategorijoms.

Pradinė duomenų analizė pradėta nuo priklausomojo kintamojo pasiskirstymo analizės, žr. 1 lentelėje. Nesunku pastebėti, kad analizuojama duomenų imtis yra subalansuota, t. y. reagavimo į iškvietimus požymis pasiskirstęs tolygiai.

1 lentelė. Duomenų rinkinio stebėjimų pasiskirstymas

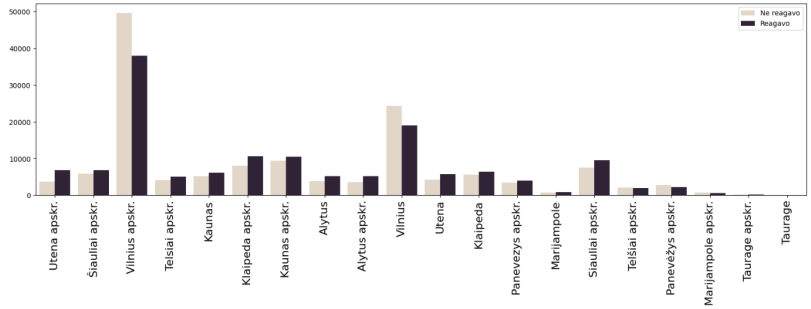
Bendras stebėjimų skaičius	Klaidingų iškvietimų skaičius	Tikrų iškvietimų skaičius
289337	144685	144652

Iškvietimų pasiskirstymo analizė Lietuvos rajonuose pateikta 1 pav. Pastebima, kad Vilnius ir Vilniaus rajonas turi žymiai daugiau fiksuojamų iškvietimų. Be to, šiuose regionuose, klaidingų iškvietimų yra užfiksuojama daugiau negu reaguotinų. Panaši tendencija taip pat pastebima Telšių, Panevėžio ir Marijampolės regionuose.

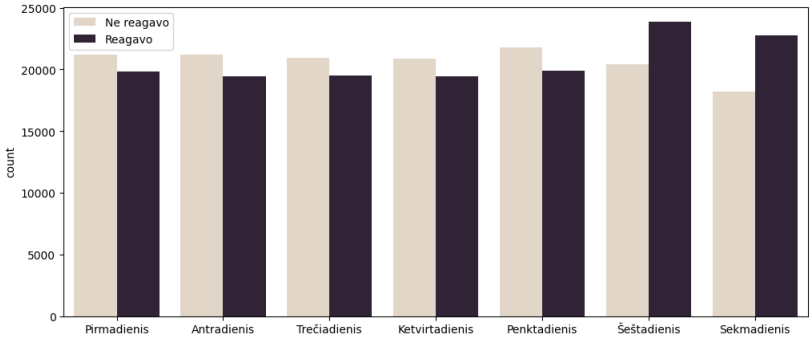
Tyrimo metu taip pat buvo analizuojama savaitės/paros laiko įtaka reagavimui į iškvietimus. Atlikus iškvietimų skaičiaus pasiskirstymo savaitėje

[1] https://maps.ird.lt/map/

analizę, kuri pateikta 2 pav., pastebėta, kad darbo dienomis, t. y. pirmadienį-penktadienį, į iškvietimus reaguojama rečiau, be to, bendras iškvietimų skaičius yra mažesnis, negu savaitgaliais. Šeštadieniais pastebimas bendras iškvietimų suaktyvėjimas, tiek klaidingų, tiek tų kuriems saugos tarnybos skiria didesnį dėmesį. Tuo tarpu sekmadieniais sumažėja klaidingų iškvietimų ir padidėja reaguotinų iškvietimų skaičius lyginant su darbo dienomis.

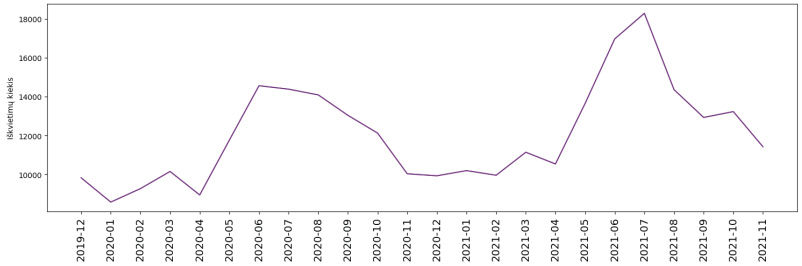


1 pav. Reagavimo į iškvietimus pasiskirstymas, pagal savivaldybes



2 pav. Reagavimo į iškvietimus pasiskirstymas, pagal savaitės dienas

Analizuojant iškvietimų dažnumą (žr. 3 pav.) pastebima, kad mažiausias iškvietimų skaičius yra fiksuojamas žiemą, pirmoje metų pusėje, t. y. sausio ir vasario mėnesiais. Pavasarį iškvietimų skaičius pradeda staigiai augti ir piką pasiekia vasarą, t. y. birželio-liepos mėnesiais, po to pradeda vėl palaipsniui mažėti. Toks tendencingumas būdingas tiek 2020 metais, tiek 2021.



3 pav. Iškvietimų dažnumas

6 Duomenų paruošimas

Tyrimo metu, buvo atlikti šie duomenų paruošimo žingsniai: skirtingų duomenų šaltinių apjungimas, naujų požymių kūrimas, treniravimo ir testavimo imties sudarymas ir požymių reikšmių normalizavimas.

Tyrime naudoti duomenys gaunami iš trijų šaltinių, buvo apjungti remiantis ilgumos ir platumos koordinatėmis. Siekiant analizę papildyti nusikalstamų veikų duomenimis kiekvienam objektui buvo priskirtas nusikalstamų veikų lygis užfiksuotas 1 km spindulio atstumu. Taip pat buvo sukurti nauji kintamieji nusakantys ar gauto iškvietimo lokacija yra miesto centre, ar iškvietimas buvo gautas šventinę dieną, ar iškvietimas buvo gautas savaitgalį, taip pat metų bei paros laiką nusakantys kintamieji, suminio iškvietimų skaičiaus kintamieji, reagavimo dažnumo kintamieji bei laiko tarpus tarp iškvietimų nusakantys kintamieji.

Ansambliniui (angl. *ensamble*) modeliui apmokyti kurtos treniravimo, testavimo ir tikrinimo (angl. validation) duomenų aibės. Treniravimo duomenų aibę sudarė 50 %, testavimo ir tikrinimo aibes po 25 % procentus duomenų rinkinio.

7 Rezultatai

Siekiant įvertinti bei palyginti klasifikavimo metodų rezultatus buvo taikomas kryžminio patikrinimo metodas su parametru *k=5* (angl. *5 fold cross-validation*). Šiuo atveju, duomenų imtis yra dalinama į penkias dalis, kur viena iš dalių yra naudojama testavimui, o likusios modelio apmokymui. Modelių galutiniai rezultatai yra šių 5 testavimo imčių vidurkiai. Gauti rezultatai pateikiami 2 lentelėje.

2 lentelė. DI modelių rezultatai gauti kryžminio patikrinimo metodu (angl. cross-validation)

Modelis	F_1	Tikslumas	Išsamumas
LogisticRegression	0,718	0,718	0,718
GaussianNB	0,682	0,691	0,674
SGDClassifier	0,726	0,744	0,723
KNeighborsClassifier	0,680	0,720	0,720
DecisionTreeClassifier	0,652	0,654	0,642
RandomForestClassifier	0,727	0,735	0,716
GradientBoostingClassifier	0,734	0,724	0,724
LGBMClassifier	0,736	0,741	0,719
XGBClassifier	**0,737**	**0,748**	**0,727**

Pastebėta, kad „*XGBoost*“ klasifikatorius pasižymi ne tik didžiausia išsamumo įverčio reikšme, tačiau ir didžiausiomis kitų tikslumo metrikų reikšmėmis. Todėl sekančiame etape pasirinkta atlikti šio modelio tinkamiausių hiperparametrų paiešką. Hiperparametrų paieškai naudoti treniravimo ir tikrinimo (angl. *validation*) duomenų aibės. Remiantis naujai gautu hiperparametrų rinkiniu ir testavimo imtimi buvo atliktas „*XGBoost*“ klasifikatoriaus testavimas, kurio rezultatai parodė, kad modelis pasiekia 0,792 teisingumą (angl. *accuracy*), tikslumą (angl. *precision*) ir išsamumą (angl. *recall*) ir F_1 įvertį, t.y. atlikta hiperparametrų paieška turėjo teigiamą įtaką rezultatų kokybei.

Siekiant padidinti klaidingų iškvietimų aptikimo modelio efektyvumą, pasirinkta apmokyti ansamblinį modelį, sudarytą iš keturių modelių, t. y. sprendimų medžio klasifikatoriaus, artimiausių kaimynų klasifikatoriaus, atraminių vektorių mašinos klasifikatoriaus ir „XGBoost“ klasifikatoriaus, su rastu geriausiu hiperparametrų rinkiniu.

Mašininio mokymosi modeliai, pasirinkti siekiant užtikrinti dvi esmines ansamblinių modelių sudarymo taisykles: modeliai turi būti skirtingi ir modeliai turi pasižymėti geru tikslumu.

Vidiniai ansamblio modeliai mokinti naudojant treniravimo duomenų aibę. Pasinaudojant tikrinimo duomenų aibe atlikti prognozavimai ir gauti procentiniai tikro iškvietimo įverčiai pateikti pagrindiniam „XGBoost“ modeliui apmokyti. Naudojantis testavimo imtimi, atlikti pilno ansamblinio modelio prognozavimai ir gauti tyrimo rezultatai.

Gauti rezultatai rodo, kad šiuo atveju pasiekiamos teisingumo (angl. *accuracy*), tikslumo (angl. *precision*), išsamumo (angl. *recall*) ir F_1 įverčio reikšmės yra lygios 0,887, t. y. sudarytas klaidingų iškvietimų aptikimo modelis pasižymi aukštu tikslumu.

8 Išvados

Tyrimo metu analizuojami saugos tarnybų duomenys siekiant sukurti klaidingų iškvietimų aptikimo metodiką. Pirmiausia buvo atliktas duomenų apjungimas, pradinė duomenų analizė, duomenų paruošimas ir modelių kūrimo eksperimentai. Skirtingų šaltinių duomenys apjungiami, siekiant gauti kiek įmanoma daugiau įžvalgų apie iškvietimo patikimumą. Duomenų analizės metu buvo nustatyti tikrų ir klaidingų iškvietimo tendencingumai, į kuriuos atsižvelgus buvo kuriami nauji požymiai.

Remiantis atliktų eksperimentų su skirtingais modeliavimo metodais rezultatais, klaidingų iškvietimų aptikimui rekomenduojama taikyti „XGBoost" klasifikatorių, kadangi šis metodas pasižymėjo didžiausiu tikslumu. Siekiant pagerinti modelio efektyvumą, buvo atlikta hiperparametrų paieška. Rezultate modelio kokybės rodikliai, apimantys teisingumo, tikslumo, išsamumo ir F_1 įverčius pagerėjo iki 0,792. Norint pasiekti aukštesnį prognozavimo tikslumą buvo sukurtas ansamblinis modelis, kurį sudarė „XGBoost", sprendimų medžių, atsitiktinių kaimynų ir atraminių vektorių klasifikatoriai. Panaudojant ansamblinio modelio rezultatus, buvo apmokytas galutinis „XGBoost" klasifikatorius, kuriuo pasiekiamas 0,886 tikslumas ir išsamumas.

Literatūra

[1] A. C. Sima, K. Stockinger, K. Affolter, M. Braschler, P. Monte, & L. Kaiser. (2018). A hybrid approach for alarm verification using stream processing, machine learning and text analytics. EDBT 2018, 26-29.

[2] M. A. A. Al-Qaness, F. Li, X. Ma, & G. Liu. (2016). Device-free home intruder detection and alarm system using wi-fi channel state information. International Journal of Future Computer and Communication, 5(4), 180.

[3] M. D. Firoozjaei, J. Park, & H. Kim. (2016). Detecting false emergency requests using callers' reporting behaviors and locations. 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, 243-247.

[4] A. M. Peco Chacon, & F. P. Garcia Marquez. (2019). False alarms management by data science. Data science and digital business, 301-316.

[5] W. Hu, T. Chen, & S. L. Shah. (2018). Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. IEEE Transactions on Industrial Electronics, 65(9), 7290-7300.

Propagandos atpažinimas lietuviškame tekste naudojant transformeriais pagrįstus, iš anksto apmokytus daugiakalbius modelius

Paulius Zaranka, Gražina Korvel

Vilniaus universitetas, Duomenų mokslo ir skaitmeninių technologijų institutas, Akademijos g. 4, LT-08412 Vilnius
paulius.zaranka@mif.vu.lt

Santrauka. Didėjant informacijos kiekiui ir jos svarbai visuomenėje atsiranda vis didesnis poreikis automatinių įrankių, gebančių atpažinti propagandą. Dėl geopolitinės situacijos Lietuvos valstybė gali būti ypatingai pažeidžiama propagandinių mechanizmų, o automatinis jos atpažinimas lietuviškuose tekstuose yra nepakankamai ištyrinėta sritis. Šio darbo tikslas – išbandyti 3 pagrindinius transformeriais pagrįstus, iš anksto apmokytus daugiakalbius modelius propagandos atpažinimui. Sprendžiamas binarinis klasifikavimo uždavinys, priskiriant tekstui propagandinio arba nepropagandinio teksto klasę. *LitLat*, *XLM-R* ir *mBERT* modeliai adaptuoti apmokant ekspertų suanotuotu duomenų rinkiniu. Nors geriausią, 88,5 % F1 statistikos įvertį pavyko pasiekti adaptavus *LitLat* iš anksto apmokytą modelį, kiti šiame darbe adaptuoti modeliai pasiekia panašius rezultatus.

Raktiniai žodžiai: propagandos atpažinimas; daugiakalbiai modeliai; transformeriai; iš anksto apmokyti modeliai; modelių adaptavimas.

1 Įvadas

Šiuolaikinėje informacijos eroje politiniai procesai pasaulyje yra stipriai veikiami propagandos [1]. Propagandiniai mechanizmai formuoja visuomenės požiūrį, elgseną ir veiksmus per sistemingą įvairios informacijos skleidimą bei sąmoningą faktų manipuliavimą. Šiais laikais internetinė erdvė yra vienas svarbiausių kanalų, per kuriuos tokia informacija plinta [2]. Nors įvairiuose kontekstuose propaganda gali turėti įvairią – tiek teigiamą, tiek neigiamą – konotaciją, dabar iš tam tikrų šaltinių sklindanti ši informacija tampa visuotiniu iššūkiu.

Šių reiškinių neigiamas poveikis visuomenėms tampa vis akivaizdesnis, todėl šiuo metu yra aktyviai vystomi tyrimai automatinio propagandos at-

DOI: https://doi.org/10.15388/LMITT.2024.22

pažinimo srityje [3][4]. Pasitelkiant natūralios kalbos apdorojimo ir giliojo mokymosi metodus kuriami modeliai, gebantys spręsti su propaganda susijusias problemas. Šis darbas tyrinėja propagandos aptikimą tekste. Pažangiausi tokio tipo tyrimai ieško būdų efektyviai spręsti uždavinį, kurio tikslas – tekste nustatyti teksto fragmentus, kuriuose yra propagandos technikų. Toks uždavinys dažnai skirstomas į du dalinius uždavinius: pirmojo uždavinio – Fragmento Identifikavimo (angl. *Span Identification*) – tikslas yra tekste atpažinti konkrečius propagandos fragmentus; antrojo uždavinio – Technikos Klasifikacijos (angl. *Technique Classification*) – tikslas – teksto fragmentui, kuris yra nustatytas kaip propagandinis, priskirti jame naudojamas technikas iš propagandos technikų sąrašo [3]. Šiame darbe išbandomi 3 pagrindiniai transformeriais pagrįsti, iš anksto apmokyti daugiakalbiai modeliai, apmokyti ir lietuvių kalba, sprendžiant paprastesnį, binarinį propagandos klasifikavimo uždavinį.

2 Transformeriais pagrįsti kalbos modeliai

Transformeriais pagrįsti kalbos modeliai jau kurį laiką dominuoja natūralios kalbos apdorojimo tyrimų srityje. Šių modelių neuroninių tinklų architektūra, leidžianti efektyviai užkoduoti ir atkoduoti žodinę informaciją [5], leido jiems pasiekti pažangiausių rezultatų daugelyje sričių, įskaitant klasifikavimą, įvardintų objektų atpažinimą, teksto generavimą ir kt. Vienas esminių šių modelių komponentų – dėmesio mechanizmas – leidžia jiems efektyviau užfiksuoti žodžių kontekstą nei ankstesnės architektūros, tokios kaip rekurentiniai neuroniniai tinklai (RNN) [5]. Spartus transformeriais pagrįstų kalbos modelių vystymasis atvedė iki tokių plačiai žinomų ir naudojamų modelių rūšių sukūrimo kaip GPT ar BERT. GPT ir kiti panašūs dideli kalbos modeliai (angl. *Large language models*) yra vienos krypties (autoregresiniai), t. y. jie generuoja tekstą nuosekliai, žodis po žodžio. Todėl, nors ir gali būti sėkmingai pritaikyti įvairioms užduotims, pagrindinis jų gebėjimas – teksto generavimas [6]. Tuo tarpu BERT yra dvikryptis modelis, kuris tekstą gali apdoroti abejomis kryptimis, t. y. iš kairės į dešinę ir iš dešinės į kairę [7]. Tai leidžia šiam ir kitiems panašiems modeliams geriau užfiksuoti kontekstą ir žodžių tarpusavio ryšius tekste, o tai daro jį tinkamesnį užduotims, kurioms reikalingas gilus konteksto supratimas. Todėl BERT šeimos modeliai yra pažangiausi sprendžiant kalbos supratimo, tame tarpe ir propagandos atpažinimo, uždavinius [8]. Įvairūs iš anksto apmokyti kalbos modeliai yra aktyviai tyrinėjami propagandos aptikimo srityje. Dideli kalbos modeliai

GPT-3 ir GPT-4 pasiekia palyginamus rezultatus [9], tačiau BERT šeimos modelis RoBERTa išlieka pažangiausias adaptavus jį propagandos atpažinimo uždaviniams spręsti [8].

Yra sukurta daugybė transformeriais pagrįstų kalbos modelių, apmokytų anglų kalba. Tuo tarpu mažiau išteklių turinčios kalbos, įskaitant lietuvių, susiduria su problemomis. Šiuo metu nėra sukurto nei vieno plačiau žinomo modelio, iš anksto apmokyto išskirtinai lietuvių kalba. Dėl šios priežasties, naudojant pažangius iš anksto apmokytus kalbos modelius spręsti lietuvių kalbos apdorojimo užduotis, dažnai yra pasitelkiami daugiakalbiai modeliai. Trys plačiausiai naudojami tokio tipo užduotims spręsti modeliai yra *LitLat* [10], *mBERT* [11, 12] ir *XLM-R* [13, 14]. *LitLat* yra apmokytas lietuvių, latvių ir anglų kalbomis; *mBERT* ir *XLM-R* – atitinkamai 106 ir 100 kalbų, įskaitant ir lietuvių. 1 lentelėje vaizduojamas bendras šių modelių palyginimas.

1 lentelė. Daugiakalbių modelių bendras palyginimas.

Modelis	Architektūra	Kalbų kiekis	Žodyno dydis
mBERT	BERT	104	119547
XLM-R	XLM-RoBERTa	100	250002
LitLat	XLM-RoBERTa	3	84201

3 Eksperimento rezultatai

Darbe nagrinėjamų modelių propagandos atpažinimo galimybėms išbandyti atliktas eksperimentas: kiekvieno jų adaptavimas ir ištestavimas sprendžiant binarinį klasifikavimo uždavinį. Adaptuoto modelio tikslas – klasifikavimas, ar duotas tekstas yra propagandinis, ar ne. Šiam uždaviniui naudotas ekspertų suanotuotas duomenų rinkinys (N = 750), nusakantis, kuriai klasei tekstas priklauso. Pagal klases lygiai subalansuotas duomenų rinkinys buvo suskirstytas į mokymo (85 %) ir testavimo (15 %) poaibius.

Kadangi BERT ir RoBERTa architektūros kaip įvestį gali priimti tik iki 512 teksto vienetų, susidedančių iš žodžių dalių, skyrybos ženklų ir specialių teksto vienetų, ilgio sekas, modeliai buvo apmokyti klasifikuoti 512 teksto vienetų ilgio gabalus. Šie gabalai sudaryti originalų tekstą skaidant slenkančio lango principu: imama 50 persidengiančių teksto vienetų iš senesnio gabalo ir 462 teksto vienetai iš originalaus gabalo. Vieną originalų tekstą vidutiniškai sudaro apie 2,73 dalys. Bendrai visiems modeliams adaptuoti naudoti parametrai pavaizduoti 2 lentelėje.

2 lentelė. Bendri modelių apmokymo hiperparametrai. Hiperparametrai buvo parinkti nepagrindžiant pasirinktų jų verčių tyrimais.

Hiperparametras	**Reikšmė**
Optimizatorius	AdamW
Mokymo žingsnis	2e-5
Paketo dydis	8
Epochų kiekis	2

Modelio testavimas vykdytas kiekvieną testavimo duomenų rinkinio tekstą suskaidžius taip pat į 512 teksto vienetų dydžio gabalus slenkančio lango principu. Kiekvienam šiam gabalui vykdyta atskira klasifikacija. Galutinė prognozė rėmėsi daugumos principu, t. y. galutinę prognozę nulemia tai, kokios klasės gabalų tekste modelis prognozavo daugiau. Rezultatai pavaizduoti 3 lentelėje. Geriausius rezultatus, 88,5 % F1 statistikos įvertį, pasiekia *LitLat* modelis. Tuo tarpu *mBERT* ir *XLM-R* modelių rezultatai tarpusavyje yra beveik identiški ir nuo geriausio modelio atsilieka nedaug – apytiksliai 2,7 %.

3 lentelė. Klasifikavimo rezultatai.

Modelis	**Tikslumas**	**Preciziškumas**	**Atkūrimas**	**F1**
mBERT	0,858	0,867	0,858	0,858
XLM-R	0,858	0,860	0,858	0,858
LitLat	**0,885**	**0,885**	**0,885**	**0,885**

4 Išvados

Šiuo metu pasaulyje automatinis propagandos aptikimas yra aktyviai tyrinėjama natūralios kalbos apdorojimo sritis. Šis susidomėjimas kyla tiek dėl jos aktualumo socialine, tiek dėl netrivialumo mašininio mokymosi prasme. Tuo tarpu panašūs uždaviniai lietuvių kalba nėra pakankamai ištirti. Šiame darbe buvo išbandyti 3 skirtingi transformeriais paremti, iš anksto apmokyti daugiakalbiai modeliai dvejetainiam propagandos klasifikavimo uždaviniui spręsti. Pagal visus pagrindinius statistikos įverčius *LitLat* modelis pasiekė geriausius rezultatus, tačiau skirtumas nuo kitų, *mBERT* ir *XLM-R*, modelių nėra ryškus. Atsižvelgiant į šiuos rezultatus visi išbandyti modeliai gali būti laikomi tinkamais tolimesniems tyrimams.

Ateityje planuojame atlikti technikų klasifikavimą ir analizuoti jų ryšį su viso teksto klasifikavimo rezultatais, kuriuos gavome šiame straipsnyje. Toks metodas leis mums geriau suprasti, kaip atskiri teksto segmentai prisideda prie bendro teksto vertinimo, bei padės išsiaiškinti, kuriuos temos dominuoja lietuviškuose, su propaganda susijusiuose, tekstuose.

Padėka

Dėkojame projektui „Propagandos ir dezinformacijos tyrimai: automatinis atpažinimas mašininio mokymo metodais, poveikis ir visuomenės atsparumas“ (Lietuvos Respublikos Vyriausybės prioritetinių mokslinių tyrimų programa (įgyvendinama per Lietuvos mokslo tarybą) „Visuomenės atsparumo stiprinimas ir krizių valdymas šiuolaikinių geopolitinių įvykių kontekste“, dotacijos numeris S-VIS-23-8) už suteiktus duomenis ir pagalbą atliekant analizę. Taip pat, esame dėkingi Vilniaus universiteto Informacinių technologijų paslaugų centrui (VU ITPC) už suteiktus didelio našumo skaičiavimo išteklius.

Literatūra

[1] Da San Martino, G., Shaar, S., Zhang, Y., Sh, Y., Barrón-Cedeno, A., & Nakov, P. (2020). Prta: A system to support the analysis of propaganda techniques in the news. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 287-293.

[2] Lietuvos nacionalinis radijas ir televizija (LRT), URL: https://www.lrt.lt/naujienos/lrt-tyrimai/5/1700792/lrt-tyrimas-lietuvos-penktoji-kolona-rusijos-propaganda-platina-seimos-gynejai-sektos-ir-knygu-apie-stalina-leidejai (žiūrėta: 2024-02-22).

[3] Martino, G., Barrón-Cedeno, A., Wachsmuth, H., Petrov, R., & Nakov, P. (2020). SemEval-2020 task 11: Detection of propaganda techniques in news articles. arXiv preprint arXiv:2009.02696.

[4] Piskorski, J., Stefanovitch, N., Da San Martino, G., & Nakov, P. (2023, July). Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023) (pp. 2343-2361).

[5] Ashish, V. (2017). Attention is all you need. Advances in neural information processing systems, 30, I.

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, Ch., McCandlish, S., Radford, A., Sutskever, I. & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[8] Abdullah, M., Altiti, O., & Obiedat, R. (2022, June). Detecting propaganda techniques in english news articles using pre-trained transformers. In 2022 13th International Conference on Information and Communication Systems (ICICS) (pp. 301-308). IEEE.
[9] Sprenkamp, K., Jones, D. G., & Zavolokina, L. (2023). Large Language Models for Propaganda Detection. arXiv preprint arXiv:2310.06422.
[10] LitLat BERT. URL: https://huggingface.co/EMBEDDIA/litlat-bert
[11] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[12] Multilingual BERT. URL: https://huggingface.co/google-bert/bert-base-multilingual-cased
[13] Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. Advances in neural information processing systems, 32.
[14] XLM-Roberta. URL: https://huggingface.co/transformers/model_doc/xlmroberta.html

Viršelio dailininkė *Jurga Tėvelienė*
Maketuotoja *Vida Vaidakavičienė*

Vilniaus universiteto leidykla
Saulėtekio al. 9, LT-10222 Vilnius
info@leidykla.vu.lt, www.leidykla.vu.lt
Knygos internete: www.knygynas.vu.lt
Mokslo periodikos žurnalai: www.zurnalai.vu.lt