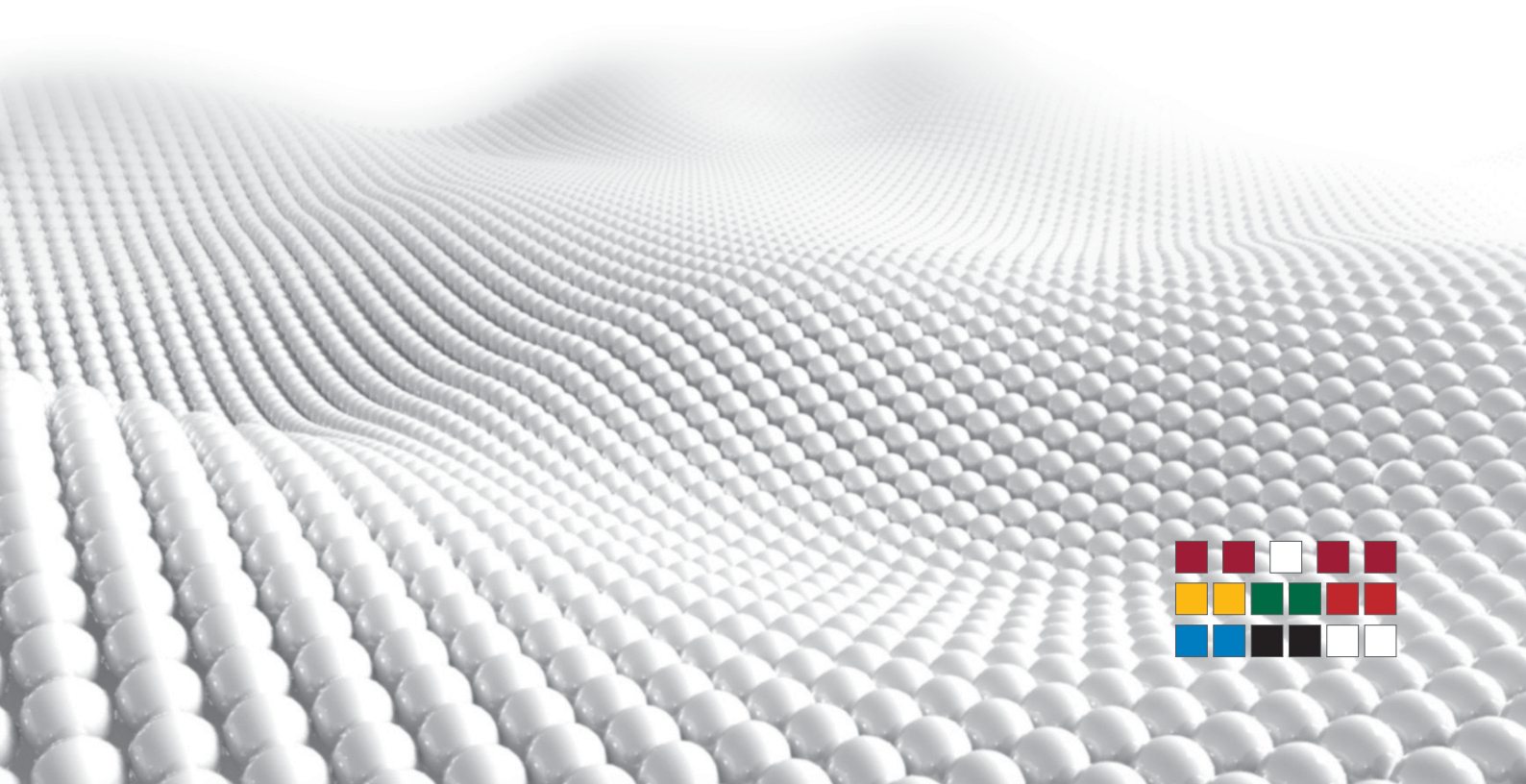
ISSN 2255-8950 (Online)
ISSN 2255-8942 (Print)

Volume 13(2025)
No. 1

Baltic Journal of Modern Computing



Co-Publishers

Vilnius University

UNIVERSITY OF LATVIA

Latvia University of Life Sciences and Technologies

Institute of Mathematics and Computer Science
University of Latvia

VIDZEMES AUGSTSKOLA

Table of Content

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 1-11
https://doi.org/10.22364/bjmc.2025.13.1.01

Evaluation of the Model for Bitcoin Price Prediction Using Machine Learning Algorithms and Blockchain Technology

Mimoza MIJOSKA, Blagoj RISTEVSKI

University St. Kliment Ohridski - Bitola, Faculty of Information and Communication Technologies, ul. Partizanska nn, 7000 Bitola, North Macedonia

mijoska.mimoza@uklo.edu.mk, blagoj.ristevski@uklo.edu.mk

ORCID 0000-0002-4248-2760, ORCID 0000-0002-8356-1203

Abstract: Blockchain technology can be used to analyze and process data through the effective integration of financial resources. Likewise, machine learning is one of the most notable technologies in recent years. Both technologies are data-driven, and therefore there is a rapidly growing interest in integrating them for more secure and efficient data sharing and analysis. This paper shows how these two technologies, blockchain technology and machine learning, can be combined to predict bitcoin volatility. To analyze and predict the volatility of bitcoin, real-time series bitcoin data was used, and the random forest algorithm was utilized. To evaluate the model, the following statistical errors were analyzed: mean absolute error, root mean square error, mean absolute percentage error, median absolute percentage error and symmetric mean absolute percentage error in cases using the different split ratios of the training and test sets. The obtained results have shown that the prediction model is well-designed.

Keywords: blockchain technology, machine learning, random forests, bitcoin volatility, statistical errors

1. Introduction

Businesses are often streamlined and enhanced by the emergence and applicability of new technological advancements. Blockchain is one of those technologies which is bringing a paradigm shift in our various old and traditional business models.

Blockchain technology was introduced in 2008 with the publication of Satoshi Nakamoto's paper - "Bitcoin: a peer-to-peer electronic cash system" (Nakamoto, 2008). Blockchain technology was first used in the cryptocurrency Bitcoin. The first Bitcoin transactions took place in January 2009. Apart from their use in the economic domain, bitcoin and blockchain technology solve an important problem in informatics and computer technology that has been an obstacle to building a functional digital monetary system for years. With this technology, the problem of double use is solved, i.e. the risk that the cryptocurrency can be used two or more times is eliminated. Virtual currency developers must prevent users from being able to spend their funds more than once. The

interest of enterprises, industries and governments around the world in blockchain technology is high, as the application of this technology is much larger than the domain of cryptocurrencies.

In 2014, a consortium called R3 was founded to start research and development of blockchain technology. In March 2017, this group counted about 75 companies, and 200 companies in March 2018, to reach 400 companies in March 2022 (Lukić, 2016) (Kramer, 2019). The formation of such a strong corporation with a lot of research and implementation of blockchain technology, especially in the financial sector, indicates that a new era in the development of banking is coming.

This paper describes the calculation of Bitcoin's realized volatility and discussion of the obtained results. The remainder of the paper is organized as follows. Section 2 highlights the principles of blockchain technology. In the next section, machine learning algorithms are described with particular emphasis on the random forest algorithm used in the research. Section 4 describes the calculation of Bitcoin's realized volatility. The discussion of the obtained results of this research is presented in the fifth section. In the last section are given concluding remarks and directions for further works.

2. Blockchain technology

"Block-chain" is a coined word made up of the words "block" and "chain". Blockchain is a distributed replicated database organized in the form of a single linked list - a chain, where nodes are blocks of transaction data. To connect the blocks, cryptographic algorithms, namely a hash function, are used in such a way that it is impossible to change the content of one block without changing the content of all the blocks that follow it. This is a very important feature of blockchain technology, as it ensures the immutability of the data entered into it. Blockchain technology enables digital transactions without intermediaries.

In 2008, several powerful financial institutions and insurance companies in the United States were on the verge of bankruptcy. These circumstances led to the immediate intervention of the federal government, to avoid a domestic and possibly global financial collapse (Senbet and Wang, 2012).

These events illustrate the dangers of living in a digital, connected world that depends on intermediaries to generate transactions and makes people vulnerable to digital exploitation and crime. It is an academic challenge to create a digital infrastructure for disbursal, without intermediaries, that has no corrupt or error-prone central authority and is secure and trustworthy. In a blockchain, ledgers are distributed across the entire network and there is no need for an intermediary to be in the middle of a transaction. The technology maintains multiple copies of data, similar to a peer-to-peer file-sharing system. Each node gets a copy of the entire database (Ashurst and Stefano, 2021).

Blockchain technology is a type of distributed ledger. Bitcoin blockchain technology uses Proof-of-Work Mining (PoW), which is the oldest publicly proven method used to achieve distributed consensus (Zhang et al., 2020).

The concept of the actual software architecture of blockchain technology is explained by breaking the concept of blockchain into two separate components - block

and chain. A block can be thought of as a data container. In the case of the Bitcoin blockchain, each block contains data (such as Bitcoin transactions), block headers, block identifiers and Merkle trees (Vujičić et al., 2018). A block is a set of data that is collected and processed to fit into it through the mining process. Each block is identified through a cryptographic hash and a time stamp. When a new block is formed, it will contain the hash of the previous block, enabling blocks to form a chronologically ordered chain from the first block ever generated in the entire blockchain (also called a "genesis block") to the newly formed block. This process is repeated over and over to develop and maintain the network (Mijoska and Ristevski, 2021).

Blockchain is a technology that is constantly evolving. The most common types of blockchains are public blockchain (Gu, 2018), private blockchain and hybrid blockchain (Samuel, 1967).

3. Machine learning algorithms

Machine learning is a scientific field that allows computers to learn without being explicitly programmed (Géron, 2017).

Well-known machine learning algorithms are the random forests, k-nearest neighbors (k-NN) algorithm, artificial neural networks, support vector machines, the Naïve Bayesian classifier, etc. Machine learning algorithms come in many forms and can be classified according to the amount and type of supervision they receive during training. There are three main categories: supervised learning (Kotsiantis et al., 2006), unsupervised learning (Géron, 2017), and reinforcement learning (Géron, 2017).

In supervised learning, the desired output for the model is already known. It is presented with only an input example and has to learn to produce the predicted output (Liu and Xiaoguang, 2021).

3.1. Random forests

The Random Forest is a supervised learning algorithm used for both regression and classification. It is among the most popular machine learning algorithms due to its high flexibility and easy implementation. Consisting of multiple decision trees, just as a forest has many trees, each tree represents one vote in most decisions. Coincidence in this algorithm is used to improve its accuracy and reduce overload, which can be a huge question for such a sophisticated algorithm. These algorithms make a decision based on a random selection of data samples and receive predictions from each tree. After that, they choose the best sustainable solution through votes. The purpose of this method is to reduce the variance of the final model. It's certainly one of the most sophisticated algorithms as it builds on the functionality of decision trees. Assuming your dataset has "*m*" features, the random forest will randomly choose *k* features where $k<m$. Now, the algorithm will compute the root node among the k features by selecting a node that has the highest information gain (Vadapalli, 2021).

After that, the algorithm splits the node into child nodes and repeats this process *n* times. Now you have a forest with *n* trees. Finally, you'll perform bootstrapping, i.e,

combine the results of all the decision trees present in your forest. Technically, it is an ensemble algorithm. The algorithm generates the individual decision trees through an attribute selection indication. Every tree relies on an independent random sample. In a classification problem, every tree votes and the most popular class is the final result. On the other hand, in a regression problem, you'll compute the average of all the tree outputs and that would be your final result (Vadapalli, 2021).

4. Realized volatility

Realized volatility is defined as an estimate of the variation in returns for an investment product over a defined period, by analyzing its historical returns. An evaluation of the degree of uncertainty and/or possible financial loss/gain from an investment in a business can be calculated using volatility/variability in the entity's share prices. The most common method of estimating variability in statistics is by calculating the standard deviation, i.e. variation in values from the mean. The realized volatility or actual volatility in the market is caused by two components - a continuous volatility component and a jump component, which influence the stock prices. Continuous volatility in a stock market is affected by intra-day trading volumes. For example, a single high-volume trade transaction can introduce a significant variation in the price of an instrument (Chauvet et al., 2010).

This paper evaluates the model that predicts the price of bitcoin. High variance intra-day data is used by analysts to estimate hourly/daily/weekly or monthly frequency levels. The resulting data can be used to estimate the volatile movement of sales. Analysts use high-variance daily data to estimate hourly/daily/weekly or monthly frequency levels. The data can then be used to estimate volatile sales movement. During the analysis, data whose frequency is 1 hour from the Gemini platform were taken (see WEB, a) and then using that data, the achieved volatility is calculated with a daily frequency. The Gemini exchange tracks and creates files for daily, hourly and minute data on the prices of the time series for the physical market for pairs, US dollar (USD) and the most popular cryptocurrencies such as bitcoin, etherium, lightcoin and others. Each file can be downloaded in .csv format. There are OHLC (Open/High/Low/Close) pricing data in each file that is updated daily. For this paper, granular hourly data are taken back to the 2015 year, for the market price of the pair of bitcoin/dollar.

Achieved volatility is measured by calculating the standard deviation from the average price of the asset over a given period. Since volatility is non-linear, the realized variance is first calculated by converting values taken from the stock market into logarithmic values and measuring the standard deviation of the log-normal returns. The achieved variance is calculated by calculating the sum of the squares of the standard deviation. The achieved volatility is calculated as the square root of the achieved variance (Yokuma and Armstrong, 1995).

To calculate the achieved volatility of bitcoin, an application was created in the programming language R (Mijoska et al., 2022).

A date sequence is then added using the seq() function, which can generate the general or regular sequences from the given inputs, defining the start and end time points with a frequency of 1 hour. In the time series "08.10.2015 13:00:00" is taken as the

starting date, and "12.01.2022 12:00:00" is taken as the end date. The price of bitcoin was taken at the close of the calculations. To enhance the accuracy of the results, the logarithmic values of the bitcoin price are calculated.

For the needs of the research, a free data set is downloaded from the website (see WEB, b) in .csv format for the last 3 years for 9 features of bitcoin. The following characteristics were used: miner revenue divided by the number of transactions, miner revenue as a percentage of transaction volume, the total estimated USD value of blockchain transactions, total USD value of block rewards and transaction fees that are paid to miners, the total number of confirmed transactions per day, the average number of transactions per block in the past 24 hours, the total value of all outgoing transactions per day, the total USD value of trading volume on major Bitcoin exchanges, and the total value in USD on all transaction fees paid to miners.

To prepare the data for machine learning it is necessary to pre-process data. Normalization is a crucial step in data pre-processing for any machine learning application and model fitting. The algorithm will be more affected by the high-end values if the data is not transformed. This means that they will probably be more accurate in predicting high values than low values. The min-max normalization method was chosen for the data used in this research (Chauvet et al., 2010).

A machine learning algorithm is used to predict the realized bitcoin volatility. The random forest algorithm is chosen, which is included in the ensemble's learning methods. Ensemble learning is a type of supervised learning technique where the basic concept is to generate several training models and then simply combine their output rules or their Hx hypothesis, construct a strong model that works very well, does not overload and also balances bias and variance Bias-Variance Tradeoff. The idea is that instead of creating a single complicated and complex model that could have a large variance that leads to overload or be too simple and have a large bias that leads to insufficient fit, many training models can be generated in the training set, which eventually combine. One such technique is the random forest, which is a common joining technique used to improve the predictive outcome of decision trees by averaging them to reduce tree variance. In this algorithm, only a random subset of m predictors is used whenever we split into a training set and a test set. The number of randomly selected variables to create each tree is the main setting parameter in random forests. Turning off some of the predictors makes sense, as the result would be that each tree uses different predictors. This implies that 2 trees generated on the same training data will have randomly different variables selected in each division so that the trees will be unrelated and independent of each other. The final result of the ensemble model is determined by counting the majority of votes from all decision trees. This concept is known as bagging. Since each decision tree takes a different set of training data as input, deviations in the original training data set do not affect the final result obtained by aggregating the decisions from each tree. Therefore, bagging as a concept reduces the variance without changing the bias of the complete ensemble (Jacobucci, 2018).

The Bitcoin volatility prediction algorithm uses the forecastML package in the R programming language. When using machine learning algorithms, the model is first generated using training data, and then the test data values are predicted. After the data preprocessing is completed, the research continues by dividing the data into a training and a test set. The data used to predict the volatility of bitcoin contain 1095 observations, starting from 14.01.2019 to 12.01.2022. In the initial analysis, the first 995 observations are taken as the training set and the remaining 100 observations are used as the test set.

The forecasting method uses three different forecasting horizons in the initial analysis. These different horizons are used to be able to predict in the short and long term, to combine the predictions in the final forecast and thus minimize the error (Aristeidou, 2020). The function randomForest(), which is used for classification and regression and also can be used for assessing proximities among data entries, is then defined with its arguments. The first step in the prediction process is to create some validation windows to perform nested cross-validation. Next, we train our model and present the predictions, residuals, and some error metrics. It is then predicted on the test set using the validation windows and the actual versus predicted values are displayed. In the beginning, the size of each forecast horizon is defined (Kumar, 2019). Horizon is an argument of the create_lagged_df() function that creates the training model and prediction dataset. This function creates a list of datasets with lagged, grouped, static and dynamic features to train a forecasting model for specific forecast horizons or to predict the future with a trained model. A horizon represents a numeric vector of one or more forecast horizons, measured in rows of data. If dates are given, a horizon of value 1 would equal 1*frequency in calendar time (Chamorro-Courtland, 2021).

5. Discussion of the obtained results

In this paper, an analysis of the results obtained with tests was made in the case when the training and testing sets are divided on a precisely determined date of the time series, and a second case when the data to be taken in the training and testing set are randomly selected with the function sample.split() from the caTools library, when predicting bitcoin market price volatility. This function is used to partition a dataset into training and testing sets for model building. Analyzes of different situations are made, in which the mean absolute error (MAE) (Dewi and Rung-Ching, 2019), root mean square error (RMSE) (Kreinovich, 2014), mean absolute percentage error (MAPE) (Willmott and Matsuura, 2005), median absolute percentage error (MDAPE) (Chai and Roland, 2014) and symmetric mean absolute percentage error (sMAPE) (Khair, 2017) is measured to properly evaluate the accuracy of the prediction model.

In this paper, an analysis of the results obtained using a different division of the training and testing sets was made, namely for the following ratios 90 %: 10 %, 80 %: 20% and 66.6 %: 33.3%.

Standard errors were examined first in the training set using validation windows. From the obtained results it can be concluded that the mean absolute error (MAE) and root mean square error (RMSE) values do not change significantly by changing the ratio of training and testing set divisions. The smallest value for MAE = 0.017 and for RMSE = 0.021 when training and test datasets are randomly selected from the given dataset.

Next, the standard errors are analyzed in the testing set, using validation windows in predicting bitcoin volatility using a different split of the training and test sets and a different choice of split method. It can be concluded that the mean absolute error (MAE) and root mean square error (RMSE) get the smallest values when the training and test sets split ratio is 90 %: 10% with a fixed split of the time series data MAE = 0.009, and RMSE = 0.011.

In the next step, the standard errors are analyzed in the training set, without using validation windows when predicting bitcoin volatility. From the obtained results, it can be concluded that the value of mean absolute error (MAE) and root mean square error (RMSE) are almost identical when using a different division of the training and testing set, for the ratio 90 %:10 %, 80 %: 20 % and 66.6 %: 33.3 %, regardless of the way of partitioning the training and testing sets with a value of 0.006 and 0.010, respectively.

Figure 1 compares the mean absolute error (MAE) and root mean square error (RMSE) using a different split of the training and test sets and a different choice of split method, in the test set without using validation windows in predicting bitcoin volatility.

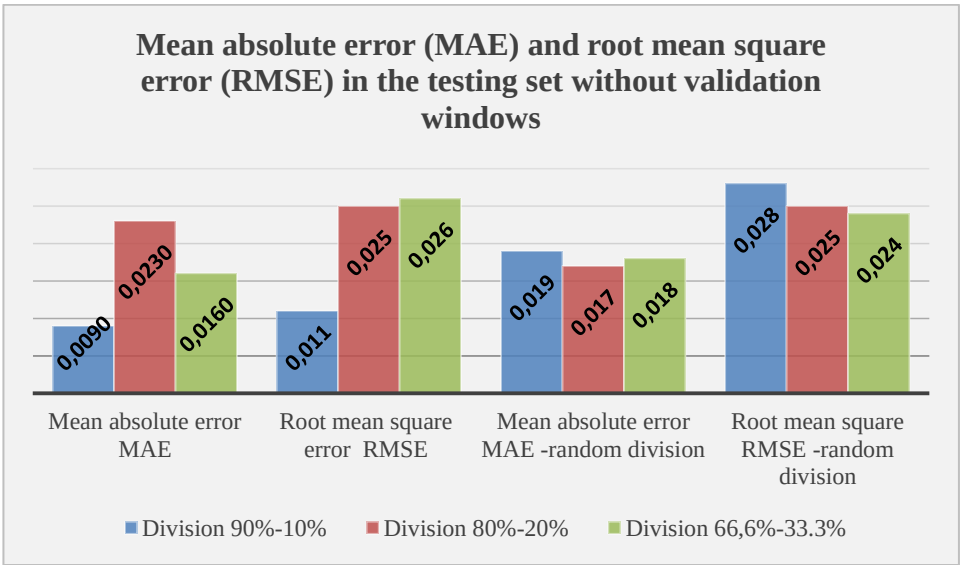


Figure 1. Mean absolute error (MAE) and root mean square error (RMSE) in the test set without validation windows in situations using a different split of the training and test sets.

From the chart in Figure 1, it can be concluded that the mean absolute error (MAE) and root mean square error (RMSE) have the smallest values when the training and test set split ratio is 90 %: 10 % with a fixed split of the time series data, MAE = 0.009 and RMSE = 0.011.

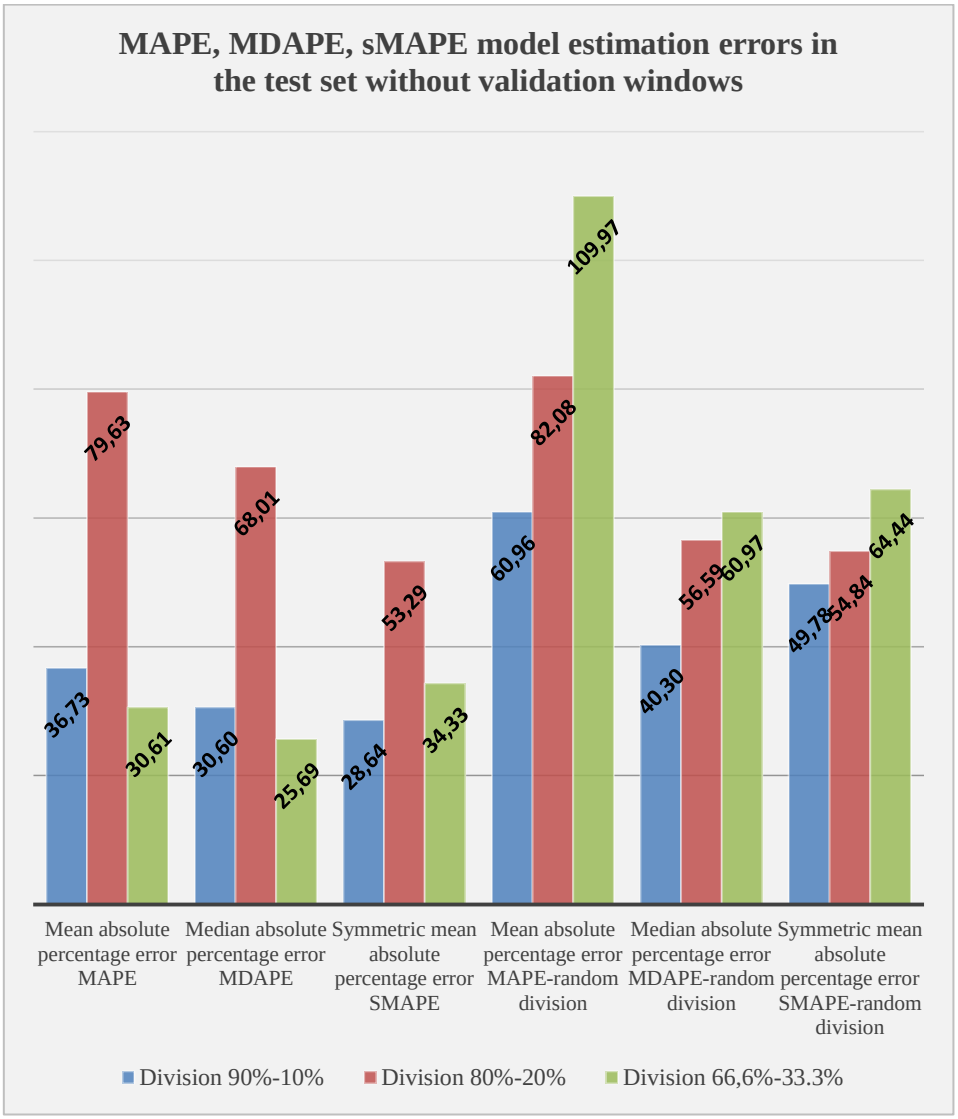


Figure 2. Mean absolute percentage error (MAPE), median absolute percentage error (MDAPE) and symmetric mean absolute percentage error (sMAPE) in situations using different partitioning of the training and test sets

From the chart shown in Figure 2, it can be concluded that the absolute percentage error (MAPE) and the median absolute percentage error (MDAPE) have significantly lower values when a fixed division of the training and testing sets is used and a selected ratio of 66.6 %: 33.3 % and thus it is concluded that this prediction model is better.

The performance metrics results are enhanced and comparable, compared with the results obtained by using two models Gated Recurrent Unit (GRU) and Bidirectional Long Short-Term Memory (BiLSTM) (Aljadani, 2022).

6. Conclusion

Blockchain technology has the potential to revolutionize the underlying payment system technology and credit information systems in banks, significantly upgrading and transforming them. The traditional system is incomplete because there is no way to prevent double-spending of money. To solve this, there is a peer-to-peer network that uses a "proof-of-work" algorithm to keep a public history of transactions. For an attacker to be able to change nodes is computationally almost impossible if honest nodes control a majority of the network.

This paper used a random forest machine learning algorithm to predict time series of realized fluctuations in the stock market price of Bitcoin and investigated whether blockchain information could be used to predict the volatility and price of Bitcoin. Many people in the world use Bitcoin as an investment because of its high volatility and in this way, they can get huge profits and losses in a short time.

In this paper, the volatility of the market price of bitcoin is modeled as a basis for measuring the risk factor in financial services using blockchain technology. Predicting the change in the value of bitcoin improves the operation of financial services, reduces the risk factor when investing, working on stock exchanges, saving, etc.

This model can also be useful for detecting anomalies and fraudulent activities in financial operations. When the actual price behavior of a cryptocurrency changes significantly from the modeled behavior, it can indicate the effect of external factors such as major global events as well as fraudulent activities. Further research could examine whether there are any macroeconomic or financial variables and indices that affect bitcoin volatility. In this paper, a specific machine learning algorithm, random forests, is chosen to predict the time series of realized volatility of Bitcoin. The same procedure can be done using another machine learning algorithm such as neural networks, support vector machines, logistic regression, lasso, k-nearest neighbor regression, etc. Additionally, one can examine which of these algorithms predicts with greater accuracy. Different types of variability can be examined as dependent variables of the model, or different types of methodology in which the prediction will not be a time series, i.e. regression, but classification where the prediction is made using an increasing or decreasing categorical variable.

In this paper, an analysis of the results obtained with tests is made in the case when the training and testing sets are divided on a precisely determined date of the time series, and a second case when the data to be taken in the training and testing set is randomly selected with the function sample.split() from the caTools package, when predicting bitcoin market price volatility. Different criteria such as forecast error measurements, the speed of calculation, interpretability and others have been used to assess the quality of forecasting. Forecast error measures or forecast accuracy are the most important in solving practical problems (Shcherbakov et al., 2013). With the analysis made in this paper and the resulting values of the standard errors, it can be said that this model can be successfully used to improve financial services, such as predicting the volatility of

Bitcoin, which is becoming more and more popular every day for people who want to invest in this cryptocurrency. Predicting the change in the value of bitcoin improves the operation of financial services, and reduces the risk factor when investing, working on stock exchanges, saving, etc.

This model can also be valuable for identifying anomalies and detecting fraudulent activities in financial operations. When the actual price behavior of a cryptocurrency changes significantly from the modeled behavior, it can indicate the effect of external factors such as major global events as well as fraudulent activities. Further research could explore whether there are any macroeconomic or financial variables and indices that influence bitcoin volatility.

References

Aljadani, A. (2022). DLCP2F: a DL-based cryptocurrency price prediction framework. *Discover Artificial Intelligence, 2(1)*, 20.

Aristeidou, C. (2020). Study of the volatility of bitcoin cryptocurrency using machine learning methods: an implementation in R, available at http://nemertes.library.upatras.gr.

Ashurst, S., and Stefano, T. (2021). Blockchain applied: practical technology and use cases of enterprise blockchain for the real world. *Productivity Press*.

Chai, T., and Roland R., D. (2014). Root mean square error (RMSE) or mean absolute error (MAE). *Geoscientific Model Development Discussions 7, no. 1*,, pp. 1525-1534.

Chamorro-Courtland, C. (2021). The Future of Clearing and Settlement in Australia: Part II-Distributed Ledger Technology. *Company & Securities Law Journal 38, no. 7*.

Chauvet, M., Senyuz, Z., Yoldas, E. (2010). What Does Realized Volatility Tell Us About Macroeconomic Fluctuations? *Unpublished working paper*.

Dewi, C., and Rung-Ching, C. (2019). Random forest and support vector machine on features selection for regression analysis. *Int. J. Innov. Comput. Inf. Control 15, no. 6*, pp. 2027-2037.

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. *O'Reilly Media, Inc*.

Gu, J. B. (2018). Consortium blockchain-based malware detection in mobile devices. *IEEE Access 6*, pp. 12118-12128.

Jacobucci, R. (2018). Decision tree stability and its effect on interpretation, available at https://osf.io/m5p2v/

Khair, U. H. (2017). Forecasting error calculation with mean absolute deviation and mean absolute percentage error. *Journal of Physics: Conference Series, vol. 930, no. 1, IOP Publish*, p. 012002.

Kotsiantis, S., Zaharakis, I., Pintelas, P. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review, 26(3)*, pp.159–190.

Kramer, M. (2019). An overview of blockchain technology based on a study of public awareness. *Global Journal of Business Research 13, no. 1*, pp. 83-91.

Kreinovich, V. H. (2014). How to estimate forecasting quality: A system-motivated derivation of symmetric mean absolute percentage error (SMAPE) and other similar characteristics, available at https://scholarworks.utep.edu.

Kumar, M. S. (2019). Credit card fraud detection using random forest algorithm. *In 3rd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 149-153.

Liu, X. and Xiaoguang, D. (2021). TanhExp: A smooth activation function with high convergence speed for lightweight neural networks. *IET Computer Vision 15, no. 2*, pp. 136-150.

Lukić, V. (2016). Potentials and limits of private digital currencies, available at http://www. ekof. bg. ac. rs/wp-content/uploads/2016/03/Seminar-katedre-2017-Potencijali-i-ograni% C4% 8Denja-privatnih-digitalnih-valuta-PDF. pdf.

Mijoska, M. and Ristevski, B. (2021). Possibilities for applying blockchain technology–a survey. *Informatica 45, no. 3*.

Mijoska, M., Ristevski, B., Savoska, S., Trajkovik, V. (2022). Predicting Bitcoin Volatility Using Machine Learning Algorithms and Blockchain Technology, available at https://repository.ukim.mk.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review: 21260*.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II—Recent progress. *IBM Journal of research and development 11.6*, pp. 601-617.

Senbet, L. W. and Wang, T. (2012). Corporate financial distress and bankruptcy: A survey. *Foundations and Trends® in Finance 5, no. 4*, pp. 243-335.

Shcherbakov, M. V., Brebels, A., Shcherbakova, N., Tyukov, A., Janovsky, T., Kamaev, V. (2013). A survey of forecast error measures. *World applied sciences journal 24, no. 24*, pp. 171-176.

Vadapalli, P. (2021). Random Forest Classifier: Overview, How Does it Work, Pros & Cons, available at https://www.upgrad.com/blog/random-forest-classifier.

Vujičić, D., Jagodić, D., Ranđić, S. (2018). Blockchain technology, bitcoin, and Ethereum: A brief overview. *17th international symposium infoteh-jahorina (infoteh)*, pp. 1-6.

WEB(a). *Gemini Exchange Data*, available at https://www.cryptodatadownload.com/data/gemini.

WEB(b). *Blockchain Charts*, available at https://www.blockchain.com/charts.

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research 30, no. 1,*, pp. 79-82.

Yokuma, J. T., and Armstrong, J. (1995). Beyond accuracy: Comparison of criteria used to select forecasting methods. *International Journal of Forecasting 11, no. 4*, pp. 591-597.

Zhang, L., Xie, Y., Zheng, Y., Xue, W., Zheng, X. (2020). The challenges and countermeasures of blockchain in finance and economics. *Systems Research and Behavioral Science 37, no. 4,*, pp. 691-698.

Received January 10, 2024, revised September 16, 2924, accepted November 11, 2024

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 12-31
https://doi.org/10.22364/bjmc.2025.13.1.02

Affective Computing for Managing Crisis Communication

Egle KLEKERE

Faculty of Science and Technology
University of Latvia, Riga, Latvia

eklekere@umich.edu

ORCID 0000-0002-2960-599X

Abstract. With affective computing being used as a tool that enhances decision-making in fields other than computing, this article exploits the potential of its applications in crisis communication. The article reviews emotion representation in different crisis communication models, leading to the identification of a research gap in these models and proposes an initial version of a Conceptual Framework for Affective Computing Supported Crisis Communication. The proposed framework underscores the significance of emotions as a pivotal factor influencing attitudes and behaviours during crises and integrates affective computing solutions aimed at effectively monitoring crises and determining suitable crisis communication strategies.

Keywords: affective computing, crisis communication, emotions in crises, computational communication

Introduction

Affective computing is a sub-field of human-computer interaction research that focuses on recognizing, analyzing, and interpreting different emotional states (Tao and Tieniu, 2005). The concept of a machines' ability to recognize, interpret and respond to human affective states was proposed in 1995 and further developed by MIT computer science professor Rosalind W. Picard (Picard, 1995; Picard, 1997; Picard et al., 2001). She was the first to suggest that machine intelligence needs to include emotional intelligence, and that computers might be given the ability to "have emotions." Since then, affective computing has grown into an interdisciplinary research area that draws from cognitive science, psychology, physiology as well as computer science to ensure that computers can identify human emotions and respond intelligently to them. Thus, affective computing is often used interchangeably with the term emotion AI (Ho et al., 2021), as it is seen as a key to advancing the development of human-centric AI.

Given that affective computing is being used as a tool to enhance decision-making in fields other than computing, this article explores affective computing as a promising approach to crisis communication research and practice.

Coombs (2009) suggests that "a crisis can be viewed as the perception of an event that threatens important expectancies of stakeholders and can impact the organization's performance. Crises are largely perceptual. If stakeholders believe there is a crisis, the organization is in a crisis unless it can successfully persuade stakeholders it is not. A crisis violates expectations; an organization has done something stakeholders feel is inappropriate." Crisis communication researchers have established a common understanding of the role of emotions. During an organizational crisis — a sudden and unexpected event that threatens to disrupt an organization's operations and poses both a financial and a reputational threat (Coombs 2007) — emotions play a significant role in shaping and re-shaping publics' perception of the situation as the conflict between the public and the organization intensifies. Emotions serve as a strong influence in how events are interpreted, perceived, and responded to as they unfold and evolve (Jin et al., 2007). Emotions influence the publics' short and long-term attitudes towards the organization(s) that stand behind the crisis, endanger organizations' reputation (Coombs, 2010), drive negative word-of-mouth (Coombs, 2022), and impact purchase decisions (Stockmeyer, 1996), including for example, product boycotts (Choi and Lin, 2009). Even though crises create a unique stakeholder group for organizations — the victims, it represents a relatively small subset of stakeholders. Non-victims, which is the group primarily analyzed by this thesis, is at least as important as the victims, as it is significantly broader and also judge organizations on how they handle crises (Coombs and Holladay, 2005). The ability to anticipate and understand the emotional reactions of different stakeholders influences organizations' effectiveness in crisis management and communication. Understanding the emotional impact of a crisis is essential for dealing quickly and effectively with the negative consequences of a crisis, including making informed choices about crisis communication strategies. An organization's reputation can be better safeguarded by crisis communication that considers the emotional responses of stakeholders and incorporates these insights into its post-crisis response planning (Coombs Holladay, 2005). Moreover, this helps to protect organizations involved in the crisis, and enhances the ability to protect the public interest. Two of the three most dominant crisis communication theories — the Situational Crisis Communication Theory (SCCT) and Integrated Crisis Mapping (ICM) model (Bukar et al., 2020) — incorporate the role of emotions. Other crisis communication models that recognize the effect of emotion on crisis development have been proposed (e.g., Lu and Huang, 2018). However, nuanced knowledge of the impact of emotions on people's reactions in crises and the effect of response strategies still requires more research.

The article reviews the representation of emotion in different crisis communication models, leading to the identification of the research gap in these models and proposing the initial version of the Conceptual Framework for Affective Computing Supported Crisis Communication, integrating affective computing solutions aimed at effectively monitoring crises and determining suitable crisis communication strategies.

The article is structured as follows: the first section introduces the concept and applications of affective computing; the second section explains the potential of affective computing to support crisis communication research. The third section discusses in detail how emotion is represented in crisis communication research, while the fourth section proposes the conceptual framework for affective computing-supported crisis communication research and practice. Finally, the last section summarizes the underlying principles and discusses outlooks, future directions, as well as possible

limitations and problems in the research of affective computing in the context of crisis communication.

1. Affective computing and its applications

At the core of affective computing research are technologies and applications that contribute to understanding the factors that influence human affective states and behavior, starting from text sentiment analysis to audio, and extending to visual and physiological-based emotion recognition (Wang et al., 2022). Research methods to measure and detect users' affective states include both laboratory and off-lab (Fortin-Cote et al., 2019), as well as mobile solutions (Politou et al., 2017), in both real and virtual environments (Marín-Morales, 2017). The rapid increase of online social media and e-commerce platforms, and vast amounts of textual data generated by users of these platforms, provide researchers with rich material for emotion analysis (Wang et al., 2022; Balaji et al., 2021). Facial expressions (Tsao and Livingstone, 2008), body gestures (Kapur et al., 2005), and speech (Batliner et al., 2011; Tuncer et al., 2020) are physical modalities, other than text, widely used to identify and analyze emotions. As the effectiveness of physical-based affect recognition may suffer from so-called social masking — a person's involuntary or deliberate concealment of their real emotions (Zhang et al., 2020), methods that are considered more objective but also more intrusive, measure physical modalities such as skin conductance, blood volume pulse, skin temperature, as well as physiological modalities such as electroencephalogram (EEG) (Alarcao and Fonseca, 2019) and electrocardiogram (ECG) (Sarkar and Etemad, 2020)).

Continuing to develop new methods, researchers have presented how parameters like mouse and keyboard inputs (Zimmermann et al., 2003), text input patterns on a smartphone (Lee et al., 2012; Ghosh et al., 2019), or steering wheel grip intensity (Oehl et al., 2007) are also reliable indicators of human emotions. The growth of affective computing has stimulated the creation of public benchmark databases, which mainly consist of unimodal (textual, audio, visual, and physiological) and multimodal databases. In turn, these commonly used databases have inspired the advancement of machine learning and deep learning techniques in the field of affective computing (Wang et al., 2022). Also, the analysis of large neurophysiological datasets is made easier with the use of machine learning techniques, and pattern classifiers can combine physiological characteristics gathered from various modalities (Yin et al., 2017).

Affective computing has grown into a fruitful area that aims to increase technological efficiency in fields such as robotics (Rattanyu et al., 2010), computer-assisted learning (Wu et al., 2015), human health, e.g., helping people with autism and facilitating their acquisition of social skills ((Blocher and Picard, 2002; Ward, 2018), depression detection (Deshpande and Rao, 2017), and telehealth (Lisetti and LeRouge, 2004). Affective computing is a useful approach for adding an emotional layer of human–environment relationships, thus enriching a variety of fields such as traffic planning, urban safety, human-centric tourism (Huang et al., 2020). In the fields of communication and marketing, affective computing has been applied to evaluate the effectiveness of communication narratives and materials (Valle-Cruz et al., 2021). A general challenge facing the field of affective computing is exploring more hybrid types of cognitive systems, where not only are computational resources and methods applied (as in the case of artificial cognitive systems) but also human specific affective processes (as in the case

of natural cognitive systems) are involved. This is crucial since contemporary interactions integrate both natural and artificial systems.

2. Potential for Affective Computing Application in Crisis Communication

The effects of emotion on crisis perception and development are still understudied. The environment in which crises emerge keeps changing, including media consumption patterns, increasing the importance of emotion-based communication in the digital era (Lu, 2017). As shown, the potential for further research on the various emotions a crisis can evoke has not been exhausted (Coombs, 2022). Furthermore, earlier scholarship on emotions in crises is of questionable value because of its methodological and theoretical limitations.

First, the theoretical limitations of earlier crisis communication theories, specifically the Situational Crisis Communication Theory and Integrated Crisis Mapping, are because they fail to consider that publics use emotional patterns of information processing rather than rational ones (Lu, 2017). Furthermore, earlier theories and research rely predominantly on the appraisal theory of emotion (Lazarus, 1991; Lazarus, 1999) the essence of which is that emotions are judgements grounded in (cognitive) appraisals of the personal significance of the surrounding environment (Ortony, 2022). Instead of an assumption that emotions are the result of a top-down process where evaluations and thoughts precede emotion and then emotions motivate behaviour, the theory of constructed emotion defines emotions as constructions of the world, not reactions to it (Barrett, 2017b). The constructivist approach explains the emergence of emotion as a bottom-up process where behaviour and bodily response precede and motivate emotion and cognition. It emphasizes the dynamic and context-dependent nature of emotions and views emotions as constructed by individuals based on their unique experiences and interpretations. Drawing on years of research in neuroscience, Barrett argues that an emotion is a brain's creation of what body sensations mean, a label a person assigns to the physiological state it senses (Barrett, 2017b). This, according to Barrett, (2017a) "makes classical appraisal theories highly doubtful, because they assume that a response derives from a stimulus that is evaluated for its meaning". The theory of constructed emotion provides insights for re-evaluation of crisis communication theories, as well as advantages for the development of an affective computing-supported crisis communication software.

The advantages of using the theory of constructed emotion in information technology are described in the context of the discipline of requirements engineering (Taveter and Iqbal, 2021), emphasizing that the theory of constructed emotion relates emotions to be constructed by software to the situations the software is meant for. The requirements that have been formulated by considering the theory of constructed emotion can be applied in designing interactive digital narratives and sociotechnical systems across a range of problem domains (Taveter and Iqbal, 2021).

Secondly, methodology, for example self-reporting surveys or media analysis, limits findings to the extent that survey participants can remember and articulate the emotions they experienced in response to a media outlet's decision to report and include emotions generated by crisis events in their agenda. Affective computing surpasses traditionally applied methods such as media content analysis, self-reported data, or even sentiment

analysis that only determines the polarity of textual data. By analyzing actual behavior and expressions of emotions, affective computing allows for relatively objective and accurate assessment, modeling, and prediction of moods, emotions, and reactions in society and its various groups. Consequently, it successfully addresses the challenges of reliability and accuracy posed by experimental, quasi-experimental, and correlational research methods. Additionally, affective computing in multi-agent communication makes a vital contribution when compared to game or rational choice theories by accurately incorporating emotions and their impact into specific situations and their models (e.g. Saunier and Jones, 2014; Peng and Su, 2020).

Thirdly, even though the relevance of computational methods has been recognized for different areas of crisis communication: organizational crises, public health crises, natural disasters, and political crises (van der Meer et al., 2022), affective computing has not been explored as an avenue for crisis communication research. The few exceptions are studies that propose the application of affective computing in the design of realistic crisis management training, incorporating emotional and stress management aspects (MacKinnon and Bacon, 2012; Mackinnon et al., 2013; Daoudi et al., 2020).

The application of deductive and inductive computational techniques in the field of communication research has been accelerated by the significant amount of data and "digital traces" left around different digital sources such as online social media platforms like Twitter, Facebook, Instagram, and TikTok as they have become a dominant means for individuals to express their thoughts and emotions about current events. Thus, computational communication science is a quickly developing sub-field among communication researchers and is characterized by the involvement of large and complex data sets — e.g., communication artifacts such as tweets, posts, emails, reviews, and other digital traces or "naturally occurring" data, as well as algorithmic solutions developed to study human communication by applying and testing communication theories (van Atteveldt and Peng, 2018). Among the computational communication approaches applicable to crisis analysis, van der Meer et al. (2022) lists deductive approaches as *dictionary methods* and supervised machine learning, as well as inductive approaches such as unsupervised methods and different cluster techniques, network analysis, distributed word embeddings, deep learning or neural network models, and machine vision. Computational communication approaches are used both for confirmatory studies to verify researcher assumptions using predefined categories for the classification of text, for example, to detect sentiment or identify communication frames, and exploratory research such as texts might be automatically classified into (potentially) meaningful categories. (van der Meer et al., 2022).

Developing and integrating affective computing approaches into computational crisis communication research would significantly widen the potential outcomes of such research as it precisely focuses on emotions and adds research modalities other than text analysis, which dominates existing research. Affective computing presents a wide range of computational approaches, e.g., analysis of speech, facial expressions, gestures, audiovisual materials, physiological characteristics, etc., to deepen the understanding of how people actually feel when facing different types of crises. Hence, affective computing has a great potential to benefit both crisis communication researchers and practitioners by building a well-grounded understanding of the effects of emotion during crises.

3. Review of Emotion Representation in Crisis Communication Models

The growing focus on the role of emotions in crisis communication has resulted in several crisis communication models and adjustments to models that previously overlooked this phenomenon. This section introduces the two dominant crisis communication theories — SCCT and ICM — that both incorporate emotions into the reasoning about the crisis outcomes and best response strategies, the model that is developed based on the critique of the previously mentioned theories — Emotion-cognition dual-factor model of crisis communication, as well as the STREMII model that describes dealing with crisis communication in social media. Finally, the section summarizes the limitations of these crisis communication theories in relation to the presence and impact of emotions in crisis.

3.1. Situational Crisis Communication Theory

SCCT is the most dominant crisis communication model and was developed by W.T. Coombs (1995), later tested, evaluated, and clarified (e.g., Coombs 2005, 2007, 2022; Coombs and Holladay 1996, 2001, 2002, 2005; Choi and Lin, 2009; Frandsen and Johansen, 2017). Conceptually focusing on “rational” aspects of cognition, SCCT incorporates affect as one of a number of crisis outcomes along with the organization's reputation and behavioral intentions like purchase intentions and negative word-of-mouth (Coombs, 2022). SCCT is built on the idea that the most effective crisis response depends on situational influences. Its foundation lies in Attribution theory, which examines the cognitive process behind attributing responsibility for events. Based on this, SCCT suggests that the response to a crisis should align with the level of responsibility stakeholders will attribute to the organization.

At the core of SCCT are the crisis types or frames that are used to interpret the crisis and crisis interventions — words and actions used in response to the crisis. Depending on responsibility attribution, crisis types include victim, accidental, and preventable crises. While victim crises are those, where the organization is perceived as a victim, accidental crises are seen as unfortunate events where the organization's responsibility is limited; preventable crises evoke strong perceptions of crisis responsibility as it is assumed that the crisis could have been prevented if the organization had taken the proper steps (Coombs and Holladay, 2002). SCCT categorizes crisis response strategies into three clusters: deny, diminish, and deal. The "deny" crisis aims to dissociate the organization from the crisis. The "diminish" group strives to minimize the organization's responsibility and the impact of the crisis. The "deal" group takes steps to assist those affected by the crisis and is seen as accepting responsibility (Coombs and Holladay, 2010).

Additionally, SCCT describes factors that alter attributions of crisis responsibility and intensify the threat from the crisis. These factors are the organization's crisis history and prior reputation (Coombs 2004; Coombs and Holladay 2001). Later developments of SCCT also add cultural aspects (Huang et al. 2016), rhetorical arena, and the "multivocal approach" to crisis communication (Frandsen and Johansen, 2017)) as contextual modifiers to crisis responsibility attribution that can increase or decrease attributions of crisis responsibility associated with the crisis type. The rhetorical arena refers to the

various voices speaking in the crisis, thus shaping attributions of crisis responsibility (Coombs, 2022).

As a result, depending on responsibility attributions, crises lead to different affective states. According to the research (Coombs and Holladay, 2005), crises from the victim-crisis cluster produced the strongest feelings of sympathy, while organizational misdeed crises produced the strongest feelings of anger and schadenfreude (the pleasure felt at someone else's misfortune (Smith, 2018)). Accident-cluster crises tended to produce muted emotional responses, whilst intentional-cluster crises generated the strongest anger. Anger, according to the research of Coombs and Halladay (2007) fuels the potentially damaging negative communication dynamic and is shown to be a mediator between crisis responsibility and negative word-of-mouth helping to convert attributions of crisis responsibility into negative word-of-mouth. Management misconduct and scansis crises (a combination of crisis and scandal) produce another emotional state caused by perceptions of unfairness and exploitation — moral outrage (Tachkova and Coombs, 2022). Based on appraisals, not attribution, moral outrage serves as a boundary condition for SCCT, where the theory's recommended crisis intervention has no effect on the common crisis outcomes of reputation.

3.2. Integrated Crisis Mapping Model (ICM)

By analyzing an audience's preferred coping strategies (problem-focused or cognitive-focused) and the level of involvement of the responsible organization, researchers predict the audience's expected emotional response — anger, sadness, anxiety, or fright (Jin et al., 2012).

ICM is derived from Lazarus's (1991, as seen in Jin et al., 2007) theory of cognitive appraisal in the field of emotion research. The authors of ICM propose the existence of two forms of coping: (a) problem-focused coping, which involves modifying the connection between the public and the organization through practical actions and steps taken; and (b) cognitive-focused coping, which involves altering only the perception of the relationship held by the public. The second aspect of the ICM model is the degree of involvement by the organization, which can range from high to low. The level of organizational involvement is determined by the relationship between the crisis events and the organization's objectives for operational and reputation success. This is based on Lazarus's primary appraisal concepts, as well as the organization's accountability for the crisis, as defined by Coombs's SCCT (Coombs, 2007). Each model's quadrant categorizations of crisis types are conceptualized based on three criteria: 1) Internal-external, 2) Personal-public, and 3) Unnatural-natural.

According to the initial ICM (Jin at el., 2007), four negative emotions - anger, fear, anxiety, and sadness - dominate public crisis situations in society. Additionally, multistage testing of the model found evidence that anxiety was the default emotion that publics felt in crises. ICM suggest that the primary audience is likely to experience two levels of emotions. The primary level of emotion represents the public's immediate reaction, while the secondary level of emotion emerges in subsequent encounters, contingent upon the organization's crisis responses. This secondary emotion might be transferred from the dominant emotion or exist alongside the primary emotional response (Jin et al., 2012).

Researchers have tested and confirmed the validity of this model by analyzing mass media publications, which are rather limited in their ability to draw full and comprehensive conclusions about the emotional reactions of the affected audiences. The need to further develop this model is demonstrated by studies analyzing people's reactions on social networks to crisis situations (Yeo et al., 2019; Varma and Perkins, 2020) It has been concluded that the list of crisis emotions defined by the ICM is not exclusive and can be refined or extended depending on the specific crisis event. This includes that the range of emotions can vary from one crisis phase to another, and that negative emotions can be accompanied by neutral or even positive ones, for instance, joy as a reaction to the important developments in a crisis (Yeo et al., 2019).

3.3. Emotion-Cognition Dual-Factor Model of Crisis Communication

Authors of the Emotion-Cognition Dual-Factor Model of Crisis Communication (EDMCC) (Lu and Huang, 2018) point out the theoretical limitations, oversimplified and unitary accounts of the cognitive process in SCCT and ICM that diminish the possibility of fully accounting for the interaction between emotion and cognition. They base their work on the assumption that the public processes crisis information in multiple stages rather than in a straightforward, unitary way. Second, depending on the intensity of the initial emotions from the crisis – defined as"the publics' cognitive appraisal of initial crisis information that gives rise to discrete crisis emotions – the model proposes that perception, evaluation, and verdicts regarding organizations during a crisis can be driven not only by cognitive but also emotional factors. According to the model, the publics' initial emotional response is shaped not only by cognitive appraisal but also by the framing effects of crisis information and the mechanism of emotional contagion. Lu and Huang explain that emotional or rational framing of the crisis event may influence perception as, initially, publics' knowledge of the crisis event is based on information released by the organization involved or the media, rather than information about what has happened. Similarly, following the actual crisis event, publics' emotions are triggered or intensified by online emotional contagion during which the publics experiences the negative emotions communicated by online forums and comments.

In contrast to SCCT and ICM, EDMCC incorporates an emotion-to-cognition approach as possible and critical for understanding the publics' evaluation of organizational crises. Lu and Huang (2018) further explain that there are four ways that initial high-intensity crisis emotions may influence how publics process crisis information: information processing routine, selective processing, information recall, and responsibility attribution.

Referring to scholars working on the relationship between cognition and emotion (Lazarus, 1999; Gordon and Arian, 2001, as seen in Lu and Huang, 2018), the authors of EDMCC explain the two-way relationship between cognition and emotion. They emphasize the significance of emotions occurring prior to succeeding thoughts while recognizing that emotions might also be responses to prior meaning and demonstrate that both emotional and logical pathways can influence decision-making.

According to the model, a significant factor that impacts whether publics will lean towards cognitive-oriented or emotion-oriented patterns is the intensity of the initial crisis emotion. If publics experience initial crisis emotions with low intensity, they will follow a cognitive-oriented pattern and may not be influenced by crisis emotions. If

publics experience the initial crisis with strong emotion, it will follow an emotion-oriented pattern, in which the effects of the initial emotion are evident in both their behavioral intentions and their cognitive processes. According to the model, individuals who have experienced initial crisis emotions with high intensity may exhibit behaviors intended to deal with the organizational crisis prior to processing subsequent crisis information, perform systematic or heuristic processing of subsequent crisis information, as well as selective processing of emotion-congruent crisis information. The intense initial crisis emotions may also promote emotion-congruent recall of crisis memories concerning the crisis-bearing organization and influence the publics' attributions of crisis responsibility and attribution approach (situational or dispositional) (Lu and Huang, 2018).

3.4. Dealing with a Crisis in the Digital Era: STREMII Model

When facing a crisis, people use social media platforms to share information — text, images, videos, or social media posts made by other users. Social media significantly changed the communication landscape by enabling dynamic, often real-time interaction and gives voice to consumers as pivotal authors of brand stories (Gensler et al., 2013). Referring to the prior research on emotions, Lu and Huang (2018) noted that intense emotions are increasingly likely to trigger behaviors in digital environments directly. This might include the desire to share online content, "share" or "like" videos, thus dramatically expanding the negative influence of organizational crises through viral forwards and negative online comments. Consequently, such activity triggers high levels of emotional intensity. By not considering emotions, organizations may fail to properly evaluate the crisis and fail in attempts at crisis communication.

On the other hand, social media platforms aid crisis managers in spreading their information in real-time and directly to the target audiences, thus providing an alternative to media framing and the agenda-setting effects on information that reaches crisis stakeholders. As SCCT and other dominant crisis communication models emerged from a mass communication model that was qualified as supporting a one-to-many communication flow and social media has changed the communication landscape significantly, the STREMII model of social media crisis communication has been proposed to fill the gap on social media effects on crisis communication (Stewart and Wilson, 2016).

The STREMII model builds on the SCCT and explains social media crisis communication as a cyclical process consisting of six elements: (1) surveillance and social listening, (2) targeting the appropriate audience, (3) responding to the crisis and conversation, (4) monitoring the landscape and evaluating outcomes, (5) interacting with consumers and publics, and (6) implementing necessary changes. Unlike SCCT, ICM and EDMCC, the STREMII model does not explain the crisis communication process and effects from the public's perspective, it rather is an actionable step by step explanation of the activities required from crisis communicators to control crisis information flow and comprehending the communication specifics in the social media environment. It does not incorporate or explain the role of emotions in crisis situations, however as the most prominent model that explains crisis communication in the context of the latest developments in the media environment (Bukar et al., 2020), it is still

important in the context of this article's ambition to propose a framework for emotional AI in crisis.

The STREMII model is consistent with the Coombs view of the crisis lifecycle as divided into three stages: pre-crisis, crisis, and post-crisis (Stewart and Wilson, 2016). The first two STREMII elements — surveillance and social listening and precise targeting are both related to the pre-crisis phase. Responding and monitoring the social media landscape (the third and fourth element of the STREMII model) are part of the active crisis phase. The post-crisis phase is associated with the fifth and sixth elements — interacting with consumers and other stakeholders and implementing necessary changes.

Revisiting the model, Stewart and Young expanded the role of the first element of the model — surveillance and social listening, which is widely known as the process of identifying and assessing what is being said about a person, brand, or business online (Jaume, 2013, as seen in Stewart and Young 2018)).

The phrase "social listening" is commonly used to describe the practice of using specific software for monitoring discussions, complaints, and trends related to specific topics or brands of significance across different social media platforms. It is done to better engage with their customers, research competitors, be able to address user complaints immediately, or even replace focus groups and surveys to determine user needs (Pomputius, 2019). Westermann and Forthmann (2021) have demonstrated how explicit and implicit experiences, which are the drivers of reputation, can be systematically recorded and analysed using social listening, thus replacing traditional reputation surveys, and expanding the possibilities to investigate reputation on a large scale.

In the revised STREMII model, social listening is not limited to detecting early signs of the potential crisis, done to prevent the crisis from erupting, or the fourth step – monitoring the social media landscape and evaluating outcomes. According to the revised model, social listening should be used in each of the practices presented by the model to ensure ongoing responsive engagement — another element that has been added to the STREMII. Responsive engagement, similar to social listening, is an activity that accompanies all six elements of the initial version of the model. Social listening involves observing stakeholders' opinions and concerns, while responsive engagement involves promoting dialogue and actively engaging with stakeholders (Stewart and Young 2018).

3.5. Research gap – from discrepancies to overlooked aspects of emotions in crisis communication theories

Different crisis communication theories have different conclusions and sometimes even contradictory views on the effects of emotion in crisis and their influence on the most effective crisis intervention, as described in previous sections. Discrepancies are the result of the underlying theories like attribution theory or appraisal theory of emotions these different models are based on, and focus these theories are willing to contribute to. All the described models overlook the most current theory of emotions – the theory of constructed emotions that is increasingly gaining support in academia.

Attribution theory, on which Coombs bases his SCCT, looks at crises primarily from the perspective of the organization's reputation in the traditional media landscape. This approach does not explore phenomena including flashbulb memories ("durable memories formed in response to strong emotional experiences" (Diamond et al, 2007))

that were first defined by Brown and Kulik (1977), initial crisis reactions, and online emotional contagion described by EDMCC. Initial crisis reactions are not necessarily related to how the public sees the company's involvement, or responsibility. For example, it might be assumed that in the case of people dying in an airplane accident, the initial emotions would be sadness and sympathy towards victims – emotions that do not depend on the airline's responsibility.

Furthermore, the STREMII model posits that communication is an ongoing, cyclical process where the situation, including the public's perception of the crisis event and involved organizations, is subject to change. Emotions, including emotions towards the responsibility-bearing organization, might evolve and transform over the course of the crisis event. An organization's response might trigger a change in primary emotions; for instance, if the response is not appropriate or does not match the public's expectations, it can trigger a negative wave of emotions. Even though the STREMII model emphasizes the evolving nature of cases of crisis, it does not explicitly incorporate or analyze emotions.

ICM overlooks that emotions might be contrasting and multi-dimensional – sadness and empathy towards victims, anger towards the responsible organization, and respect for institutions that solve the issue, e.g., firefighters. Finally, EDMCC, a framework that integrates emotional factors into the processing and analysis of crisis information, includes some questionable assertions. For instance, it emphasizes a clear distinction between cognition-oriented and emotion-oriented patterns in crisis communication, which contradicts the theory of constructed emotion and its supporting evidence that emotions and cognition mutually support each other, operating in tandem. Similarly, the assertion that individuals initially experiencing crisis emotions with low intensity will adopt a cognitive-oriented pattern, thereby remaining unaffected by crisis emotions, and vice versa, is equally dubious.

4. Conceptual Framework for Affective Computing Supported Crisis Communication

The affective computing methods and techniques described above have the potential to deepen the understanding of the effects of emotion on crisis perception and development, as well as contribute to crisis communication practice (see Figure 1). Affective computing applications in lab settings, used to evaluate the emotional reactions of participants exposed to crisis-related stimulus could contribute to the debate on emotions in crisis. For instance, such experiments might provide further insights into assumptions presented by EDMCC, explaining how the intensity of the initial crisis emotions, framing effects and influences of emotional contagion have an impact on how the public perceives, evaluates, and makes verdicts about organizations during a crisis. Affective computing methods would be beneficial in testing different emotion-based crisis intervention and messaging strategies (such as emotion mirroring or empathy) and evaluating how their effectiveness is affected by the type of specific crisis case based on attribution theory – the victim, accidental or intentional crisis as defined by Coombs.

Moreover, as crisis communication models are mainly based on the appraisal theory and that is challenged by the theory of constructed emotions, the relevance of these

theories for analyzing the emotion aspects in crisis situations can be tested by applying research methods rooted in affective computing. As the theory of constructed emotion acknowledges that the body's response is the driver and motivator of emotions and cognition, affective computing methods that assess physical measures such as skin conductance, blood volume pulse, skin temperature, as well as physiological modalities like electroencephalogram (EEG) and electrocardiogram (ECG) in combination with self–defined emotional states might provide novel knowledge about crisis emotions development and effects. Constructivists emphasise the highly individualised, subjective and context–dependent nature of emotional experience, so affective computing experiments have the potential to deepen understanding of the contextual background of crisis situations and explain how this affects the formation of emotional responses.

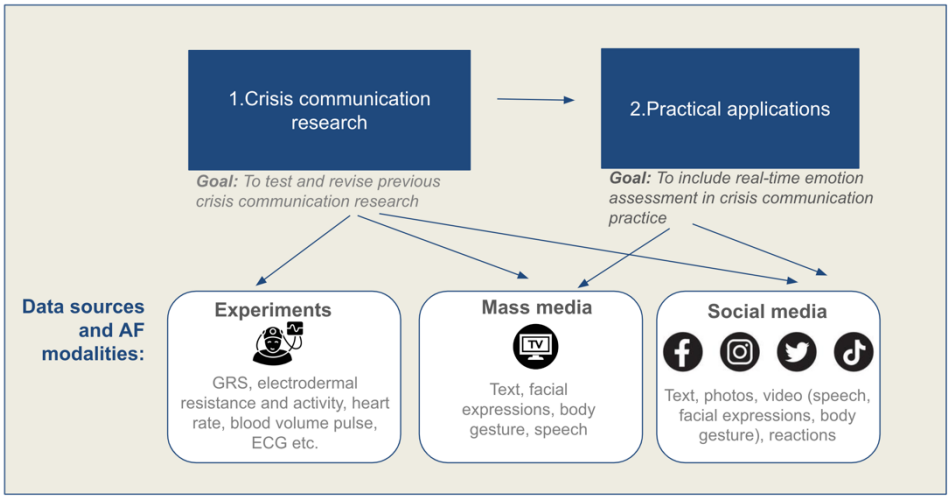


Figure 1. Applications of Affective Computing Methods for Crisis Communication research and practice

While crisis communication research could benefit from affective computing experiments in laboratory settings and conventional and social media analysis, for practical application, only less intrusive approaches would be successful. With the goal to include real-time emotion assessment to ensure more effective and precise crisis communication, affective computing methods can be applied to social networks and mass media analysis. Social media posts provide rich material for emotion analysis. That material includes texts, photos and videos of facial expressions, body gestures, speech patterns of affected people, and social media reactions that are used to signal the attitudes towards social media posts. Such information contributes to the deeper understanding of emotions related to crisis events and might be used in the calculations involved in the decision-making process in choosing the most appropriate crisis response.

To describe the potential of the application of affective computing methods in crisis communication, the Conceptual Framework for Affective Computing-Supported Crisis

Communication (ACSCC) is proposed (see Figure 2). It combines and builds on the cyclical, process-oriented approach of STREMMI and crisis phases (pre-crisis, crisis, and post-crisis) defined by Coombs. However, it redefines the activities in each crisis phase, incorporating a new concept – affective social listening – as an ongoing process that supports each crisis communication step. By incorporating emotion analysis, affective social listening evolves the concept of social listening, which is an integral part of STREMII. Affective social listening, thus, is defined as using affective computing methods for monitoring information, discussions, and trends related to topics across social media platforms to detect and interpret the public's emotions.

The proposed ACSCC defines two phases (development and application phases) and three steps of crisis communication (pre-crisis, crisis, and post crisis). The first phase corresponds to the development phase where computing research is applied to gather information necessary for the development of an affective social listening tool dedicated to crisis communication. The second phase explains the practical application of the affective social listening tool and its focus on each step in crisis communication.

To support the first step of the proposed framework (apply affective social listening to crisis risk detection), it would be necessary to analyze which emotions are associated with an emerging crisis and what social media activities, for instance, emotion words used in social media posts signal this emerging condition. The importance of emotion words has been emphasized by the theory of constructed emotions that suggests that emotion words provide an important context in emotion perception (Gendron, et al, 2012). According to this theory, conceptual knowledge about emotion, anchored with emotion words plays a key role in generating the perception of emotion.

Applying affective social listening to stakeholder segmentation as part of preparing an organization for potential crises would require empirical analysis of the connection between individual personality traits, emotional states expressed in social media, and their reactions and behaviors in crises. Affective intelligence theory analyzes such connections in the context of political communication (Marcus et al., 2011) and would be a feasible guide for similar analysis as connected to crisis communication.

To put the second step – identifying crisis emotions – into practice, crisis case studies applying affective computing methods are necessary to analyze conventional media and social media content to detect and define interconnections between specific crisis types and scenarios and the public’s expressions of emotion. Such studies would also provide insights into whether and how the intensity of the initial crisis emotions, framing effects, and influences of emotional contagion have an impact on society's perception, evaluation, and verdict on organizations during a crisis. Appropriate crisis intervention requires testing different emotion-based crisis intervention and messaging strategies (such as, for instance, emotional versus rational framing of crisis response (Claeys and Cauberghe, 2014), demonstration of shame and regret (van der Meer and Verhoeven 2014 etc.) and evaluating how their effectiveness is affected by the type of specific crisis. Gathering and analyzing such information sets the ground for building an emotion-aware information technology tool that supports crisis managers in decision making regarding crisis communication.

The third step – post-crisis reputation evaluation and learning from crises resulting in implementing necessary changes – need to be detailed by applying computational reputation measurement and stakeholder analysis solutions similar to that presented by Westermann and Forthmann (2021).

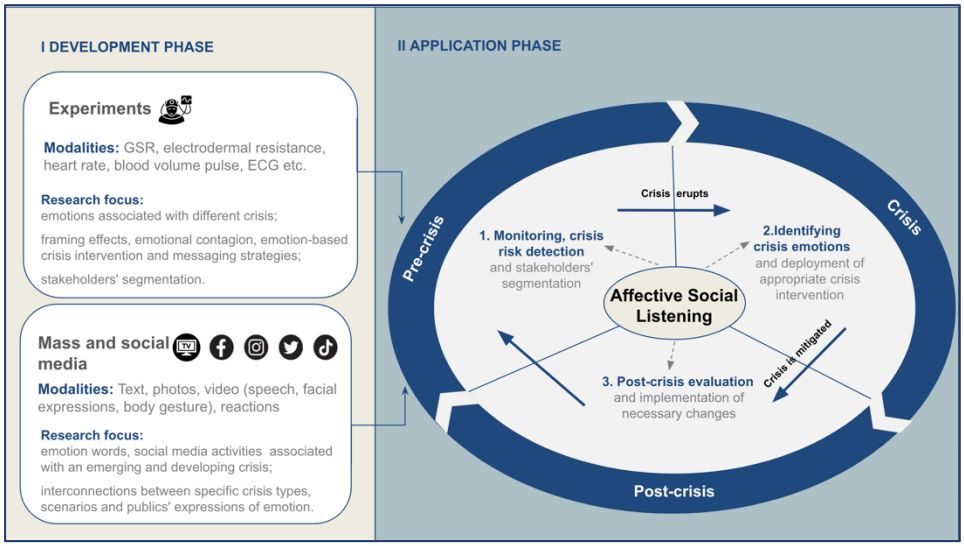


Figure 2. Conceptual Framework for Affective Computing Supported Crisis Communication (ACSCC)

All three ACSCC crisis communication steps in the application phase are supported by affective social listening that informs crisis communication decision-making. The first – monitoring and crisis risk detection – is related to the pre-crisis phase where affective social listening would contribute to an organization's ability to identify issues early and provide timely response to the potential crisis. In the pre-crisis phase, social affective computing can also be applied to stakeholder segmentation, thus preparing an organization for effective communication in case a crisis erupts. Identifying crisis emotion is part of the active crisis phase where affective social listening informs about initial crisis emotions and allows an organization to detect and address stakeholder concerns and adjust responses according to the development of their emotional reactions. Affective social listening provides the opportunity to evaluate the effectiveness of crisis intervention and adjust it accordingly. The final stage – the post-crisis phase – is focused on the post-crisis reputation evaluation and learning from crises that results in implementing changes that are necessary.

ACSCC may increase the efficiency of crisis communication by acknowledging the importance of emotions in crisis communication. ACSCC supports emotions as a factor driving attitudes and behaviors and influencing crisis communication strategies. It assumes that the ability to monitor and address society's emotional states in real-time, combined with machine learning solutions that support crisis communicators with predicting potential outcomes in different crisis scenarios, would make crisis communication more efficient, resulting in achieving intended emotional states, behaviors and mitigating reputation risks. As previously demonstrated (Iqbal, et al, 2023), the discipline of requirements-engineering combined with the theory of constructed emotion would be applied to the development of such emotion-aware technology that has the ability to influence the emotional states of the public.

5. Conclusions

The article reviews emotion representation in different crisis communication models, leading to the identification of the research gap in these models and proposing an initial version of the Conceptual Framework for Affective Computing Supported Crisis Communication. Affective computing methods and techniques have the potential to contribute to crisis communication theory and practice by deepening the understanding the effects of emotion during crises and providing options for operationalizing research tools and methods in the field of affective computing in crisis communication. Experiments involving affective computing methods would add valuable empirical material to the theoretical debate on emotions in crisis, particularly in understanding the evolving nature of emotions over the course of a crisis event. Furthermore, in real-life crises, the ability to monitor and address the public's emotional states in real-time, combined with machine learning solutions that support crisis communicators with prognoses of potential outcomes in different crisis scenarios, provides opportunities to increase the efficiency of crisis communication. This could mitigate reputation risks and assure that the intended behaviors are achieved.

The conceptual framework for affective computing-supported crisis communication (ACSCC) considers crisis communication theories and previous research on affective computing. It incorporates emotion analysis into the cyclical, process-oriented approach of crisis communication, acknowledging emotions as a factor driving attitudes and behaviors and influencing crisis communication strategies. The three steps of the ACSCC crisis communication process are supported by affective social listening – the author's proposed concept that evolves from social listening and is defined as an ongoing process that supports communication management by using affective computing methods for monitoring the information, discussions, and trends related to specific subjects across social media platforms to detect and interpret the public's emotions. The role of affective social listening is to inform crisis communication decision-making by monitoring and interpreting the public's emotions in real time.

As the proposed framework currently provides a conceptual architecture of a combined collection of methodological approaches, it requires empirical tests for its validation and further practical application.

The full-scale application of affective computing methods to support real-time decision-making amid crisis situations is limited by its intrusive nature and ethical considerations. First, intrusive affective computing methods that measure physical modalities such as skin conductance, blood volume pulse, and skin temperature, as well as physiological modalities such as electroencephalogram or electrocardiogram, are only applicable in laboratory settings and only indirectly contribute to real-life applications. Second, all the data that might provide insights into emotions generated by crisis events must be gathered ethically with the consent of analyzed individuals. Parameters such as facial expressions captured by phone or computer camera, data provided by smartwatches, mouse and keyboard inputs, and text input patterns on a smartphone may all be valuable data sources; however, their use is in question as users might not want to share such data freely and willingly. Thus, the affective computing methods applied for real-time crisis emotion analysis might be limited to publicly available data sources such as conventional and social media content. However, the current approach assumes that complementary micro-level (experimental) and macro-level (large-scale digital social

network) resources might provide a necessary insight for an accurate affective computing-based crisis management tool.

References

Alarcao, S.M., Fonseca, M.J. (2019). Emotions recognition using EEG signals: a survey, *IEEE Trans. Affect. Comput*., **10**, 374–393, https://ieeexplore.ieee.org/document/7946165.

Balaji, T.K., Annavarapu, C. S. R., Bablani, A. (2021). Machine learning algorithms for social media analysis: A survey. *Computer Science Review*, **40**, 100395.

Barrett, L.F. (2017a). The theory of constructed emotion: an active inference account of interoception and categorization, *Social Cognitive and Affective Neuroscience*, **12** (1), 1–23, https://doi.org/10.1093/scan/nsw154

Barrett, L. F. (2017b). *How emotions are made: The secret life of the brain*. Houghton Mifflin Harcourt, Boston, Massachusetts, 425 pp.

Batliner, A., Schuller, B., Seppi, D., Steidl, S., Devillers, L., Vidrascu, L., Vogt, T., Aharonson, V., Amir, N. (2011). The automatic recognition of emotions in speech, *Emotion-Oriented Systems*, **2**, 71-99.

Blocher, K., Picard, R.W. (2002). Affective social quest: emotion recognition therapy for autistic children. In: Dautenhahn, K., Bond, A.H., Canamero, L., Edmonds, B. (Eds.), *Socially Intelligent Agents: Creating Relationships with Computers and Robots*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Brown, R., Kulik, J. (1977). Flashbulb memories. *Cognition.* **5** (1): 73–99. doi:10.1016/0010-0277(77)90018-X. S2CID 53195074.

Bukar, U. A., Jabar, M. A., Sidi F., Nor, R. N. H. B., Abdullah S., Othman M. (2020). Crisis Informatics in the Context of Social Media Crisis Communication: Theoretical Models, Taxonomy, and Open Issues, *IEEE Access*, vol. **8**, pp. 185842-185869, doi: 10.1109/ACCESS.2020.3030184.

Claeys A.S., CauberghеV. (2014), What makes crisis response strategies work? The impact of crisis involvement and message framing, *Journal of Business Research*, **67**(2), 182-189.

Coombs, W. T. (1995). Choosing the Right Words: The Development of Guidelines for the Selection of the "Appropriate" Crisis-Response Strategies. *Management Communication Quarterly*, **8**(4), 447–476. https://doi.org/10.1177/0893318995008004003

Coombs, W. T., Holladay, S. J. (2005). Exploratory study of stakeholder emotions: Affect and crisis. In N. M. Ashkanasy, W. J. Zerbe, C. E. J. Hartel (Eds.). *Research on emotion in organizations: Volume 1: The effect of affect in organizational settings* (pp. 271–288). New York: Elsevier.

Coombs, W.T. (2007). Protecting Organization Reputations During a Crisis: The Development and Application of Situational Crisis Communication Theory. *Corp Reputation Rev* **10**, 163–176. https://doi.org/10.1057/palgrave.crr.1550049

Coombs, W. T. (2010). Parameters for Crisis Communication. *The Handbook of Crisis Communication*, 17–53. https://doi.org/10.1002/9781444314885.CH1

Coombs, W.T. (2022). Situational Crisis Communication Theory (SCCT). In *The Handbook of Crisis Communication* (eds W.T. Coombs and S.J. Holladay). https://doi.org/10.1002/9781119678953.ch14

Coombs W. T., Holladay S.J. (1996). Communication and Attributions in a Crisis: An Experimental Study in Crisis Communication, *Journal of Public Relations Research*, **8**:4, 279-295, DOI: 10.1207/s1532754xjprr0804_04

Coombs W. T., Holladay S.J. (2001). An Extended Examination of the Crisis Situations: A Fusion of the Relational Management and Symbolic Approaches, *Journal of Public Relations Research*, **13**:4, 321-340, DOI: 10.1207/S1532754XJPRR1304_03

Coombs, W. T., Holladay, S. J. (2002). Helping Crisis Managers Protect Reputational Assets: Initial Tests of the Situational Crisis Communication Theory. *Management Communication Quarterly*, **16**(2), 165–186. https://doi.org/10.1177/089331802237233

Coombs W. T., Holladay S.J. (2005). An Exploratory Study of Stakeholder Emotions: Affect and Crises, Ashkanasy, N.M., Zerbe, W.J. and Härtel, C.E.J. (Ed.) *The Effect of Affect in Organizational Settings (Research on Emotion in Organizations*, Vol. 1, Emerald Group Publishing Limited, Bingley, pp. 263-280. https://doi.org/10.1016/S1746-9791(05)01111-92005

Coombs, W. T., Holladay, S. J. (2010). *PR strategy and application: Managing influence*. Wiley-Blackwell.

Choi, Y., Lin, Y.-H. (2009). Consumer Responses to Mattel Product Recalls Posted on Online Bulletin Boards: Exploring Two Types of Emotion. *Journal of Public Relations Research*, **21**(2), 198–207. doi:10.1080/10627260802557506

Daoudi, I., Tranvouez, E., Chebil, R., Espinasse, B., Chaari, W.L. (2020). An EDM-based Multimodal Method for Assessing Learners' Affective States in Collaborative Crisis Management Serious Games, 13th International Conference on Educational Data Mining, online, France. ⟨hal-02961310⟩

Deshpande, M., Rao, V., (2017). Depression detection using emotion artificial intelligence, International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2017, pp. 858-862, doi: 10.1109/ISS1.2017.8389299.

Diamond, D. M., Campbell, A. M., Park, C. R., Halonen, J., Zoladz, P. R. (2007). The temporal dynamics model of emotional memory processing: a synthesis on the neurobiological basis of stress-induced amnesia, flashbulb and traumatic memories, and the Yerkes-Dodson law. *Neural plasticity*, 60803. https://doi.org/10.1155/2007/60803

Fortin-Cote, A., Beaudin-Gagnon, N., Campeau-Lecours, A., Tremblay, S., Jackson, P. L. (2019). Affective Computing Out-of-The-Lab: The Cost of Low Cost. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC).

Frandsen, F., Johansen, W. (2017). *Organizational crisis communication: A multivocal approach*. London: Sage. 280 pp.

Gendron, M., Lindquist, K. A., Barsalou, L., Barrett, L. F. (2012). Emotion words shape emotion percepts. *Emotion*, **12**(2), 314.

Gensler, S.,Völckner, F., Liu-Thompkins, Y., Wiertz, C. (2013). Managing Brands in the Social Media Environment. *Journal of Interactive Marketing*, **27**, 242–256. 10.1016/j.intmar.2013.09.004.

Ghosh, S., Hiware, K., Ganguly, N., Mitra, B., De, P. (2019). Emotion Detection from Touch Interactions during Text Entry on Smartphones. *International Journal of Human-Computer Studies*, Vol. **130**, 47-57

Gordon, C., Arian, A. (2001). Threat and decision making. *Journal of Conflict Resolution*, **45**(2), 196–215.

Hamborg, F.. (2022). Towards Automated Frame Analysis: Natural Language Processing Techniques to Reveal Media Bias in News Articles. http://nbn-resolving.de/urn:nbn:de:bsz:352-2-k1jb5j3vd9oq4

Ho, M., Mantello, P., Nguyen, H.K., Vuong, Q. (2021). Affective computing scholarship and the rise of China: a view from 25 years of bibliometric data. *Humanities and Social Sciences Communications*, **8**, 1-14.

Huang Y, Fei T, Kwan M-P, Kang Y, Li J, Li Y, Li X, Bian M. (2020) GIS-Based Emotional Computing: A Review of Quantitative Approaches to Measure the Emotion Layer of Human–Environment Relationships. *ISPRS International Journal of Geo-Information*, **9**(9):551. https://doi.org/10.3390/ijgi9090551

Huang, Y., Wu, F., Cheng, Y. (2015). Crisis communication in context: Cultural and political influences underpinning Chinese public relations practice. *Public Relations Review*, **42**(1): 201–213.

Iqbal, T., Marshall, J. G., Taveter, K., Schmidt, A. (2023). Theory of constructed emotion meets RE: An industrial case study. *Journal of Systems and Software*, ***197***, 111544.

Jin, Y., Pang, A., Cameron, G. T. (2007). Integrated crisis mapping: Towards a publics-based, emotion-driven conceptualization in crisis communication. *Sphera Publica*, **7**, 81–96. https://ink.library.smu.edu.sg/lkcsb_research/6034

Jin, Y., Pang, A., Cameron, G.T., (2012). Toward a Publics-Driven, Emotion-Based Conceptualization in Crisis Communication: Unearthing Dominant Emotions in Multi-Staged Testing of the Integrated Crisis Mapping (ICM) Model, *Journal of Public Relations Research*, **24**:3, 266-298, DOI: 10.1080/1062726X.2012.676747

Yin, Z., Zhao, M., Wang, Y., Yang, J., Zhang, J. (2017). Recognition of emotions using multimodal physiological signals and an ensemble deep learning model. *Computer methods and programs in biomedicine*, **140**, 93-110.

Kapur, A., Kapur, A., Virji-Babul, N., Tzanetakis, G., Driessen, P.F. (2005). Gesture-Based Affective Computing on Motion Capture Data. In: Tao, J., Tan, T., Picard, R.W. (eds) *Affective Computing and Intelligent Interaction*. ACII 2005. Lecture Notes in Computer Science, vol **3784**. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11573548_1

Lazarus, R. S. (1991). *Emotion and adaption*. New York, NY: Oxford University Press.

Lazarus, R. S. (1999). The cognition-emotion debate: A bit of history. In T. Dalgleish, M. J. Power (Eds.). *Handbook of cognition and emotion* (pp. 3–19). Chichester, UK: Wiley.

Lee, H., Choi, Y. S., Lee, S., Park, I. P. (2012), Towards Unobtrusive Emotion Recognition for Affective Social Communication. In: 9th Annual IEEE Consumer Communications and Networking Conference, pp. 260- 264.

Lisetti, C., LeRouge, C. (2004). Affective computing in tele-home health. 37th Annual Hawaii International Conference on System Sciences. Proceedings of the Big Island, HI, USA, pp. 8 pp.-, doi: 10.1109/HICSS.2004.1265373.

Lu, Y., Huang, Y-H. C. (2018). Getting emotional: An emotion-cognition dual-factor model of crisis communication. *Public Relations Review*, **44**(1), 98-107. https://doi.org/10.1016/j.pubrev.2017.09.007

MacKinnon, L.M., Bacon, L. (2012) "Developing Realistic Crisis Management Training," in ISCRAM.

Mackinnon, L.M., Bacon, L., Cortellessa, G., Cesta, A. (2013). Using Emotional Intelligence in Training Crisis Managers: The Pandora Approach. *International Journal of Distance Education Technologies* (IJDET), 11(2), 66-95.

Marín-Morales, J., Llinares, C., Guixeres, J., Alcañiz, M. (2020). Emotion Recognition in Immersive Virtual Reality: From Statistics to Affective Computing. *Sensors*, **20**(18), 5163.

Marcus, G., MacKuen, M., Neuman, W. R. (2011). Parsimony and Complexity: Developing and Testing Theories of Affective Intelligence. *Political Psychology*. **32**. 323 - 336. 10.1111/j.1467-9221.2010.00806.x.

Oehl, M., Siebert, F.W., Tews, T.-K., Höger, R., Pfister, H.-R. (2007). Improving human-machine interaction—a noninvasive approach to detect emotions in car drivers. In: Jacko, J.A. (Ed.), *Human-Computer Interaction, Part III*, HCII 2011, LNCS 6763. Springer-Verlag, Berlin Heidelberg, pp. 577–585.

Ortony, A. (2022). Are All "Basic Emotions" Emotions? A Problem for the (Basic) Emotions Construct. *Perspectives on Psychological Science*, **17**(1), 41-61. https://doi.org/10.1177/1745691620985415

Peng, C., Su, C. (2020). A multi- agent affective interactive MAGDM approach and its applications. *Expert Systems, 37*.

Picard, R.W. (1995). Affective Computing, MIT Media Laboratory Perceptual Computing Section Technical Report #321

Picard, R.W.(1997). *Affective Computing*, MIT Press, Cambridge, MA, USA.

Picard, R.W., Vyzas E., Healey J. (2001). Toward machine emotional intelligence: analysis of affective physiological state, *IEEE Trans. Pattern Anal. Mach. Intell*. **23**, 1175–1191 https://doi.org/10.1109/34.954607.

Politou, E., Alepis, E., Patsakis, C. (2017). A survey on mobile affective computing. *Computer Science Review*, **25**, 79–100. doi:10.1016/j.cosrev.2017.03.001

Pomputius, A. (2019) Can You Hear Me Now? Social Listening as a Strategy for Understanding User Needs, *Medical Reference Services Quarterly*, **38**:2, 181-186, DOI: 10.1080/02763869.2019.1588042

Rattanyu, K.; Ohkura, M.; Mizukawa, M. Emotion Monitoring from Physiological Signals for Service Robots in the Living Space. In Proceedings of the ICCAS 2010, Gyeonggi-do, Korea, 27–30 October 2010; pp. 580–583.

Sarkar, P., Etemad, A., (2020). Self-supervised ECG representation learning for emotion recognition, *IEEE Trans. Affect. Comput*., 1, DOI: 10.1109/ TAFFC.2020.3014842. –1.

Saunier, J., Jones, H. (2014, May). Mixed agent/social dynamics for emotion computation. In *AAMAS* (pp. 645-652).

Smith, T. W. (2018). *Schadenfreude: The joy of another's misfortune*. Little, Brown Spark.

Stockmyer, J. (1996). Brands in crisis: Consumer help for deserving victims. *Advances in Consumer Research*, **23**: 429–435.

Stewart, M. C., Wilson, B. G. (2016), The dynamic role of social media during hurricane #sandy: An introduction of the STREMII model to weather the storm of the crisis lifecycle, *Comput. Hum. Behav.*, vol. **54**, pp. 639-646

Stewart, M. C., Young, C. (2018). Revisiting STREMII: Social media crisis communication during Hurricane Matthew. *Journal of International Crisis and Risk Communication Research*, **1**(2), 279–301. https://search.informit.org/doi/10.3316/INFORMIT.096867150007700

Tachkova, E. R., Coombs, W. T. (2022). *Communicating in Extreme crises: Lessons from the Edge.* Routledge.

Tsao, D. Y., Livingstone, M. S. (2008). Mechanisms of Face Perception. *Annual Review of Neuroscience*, **31(**1), 411–437.

Tao, J., Tieniu, T., (2005). Affective Computing: A Review. Affective Computing and Intelligent Interaction. Vol. LNCS 3784. Springer

Taveter, K., Iqbal, T. (2021). Theory of constructed emotion meets RE. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pp. 383-386. IEEE.

Tuncer, T., Dogan, S., Acharya, U. R. (2020). Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques. Knowledge-Based Systems, 106547.

Yeo, S. L., Pang, A., Cheong, M., Yeo, J. Q. (2019). Emotions in Social Media: An Analysis of Tweet Responses to MH370 Search Suspension Announcement. *International Journal of Business Communication*, 232948841988275.

Varma, T. M., Perkins, S. C. (2020). Nestle India in a soup: Mapping emotions to the use of coping strategies. *Journal of Marketing Development & Competitiveness*, **14**(4), 101-115.

Valle-Cruz, D., Lopez-Chau, A., Sandoval-Almazan, R. (2021). How much do Twitter posts affect voters? Analysis of the multi-emotional charge with affective computing in political campaigns. DG. O2021: The 22nd Annual International Conference on Digital Government Research.

van der Meer, T.G.L.A., Kroon, A.C. (2022). Crisis Communication and Computational Methods. In *The Handbook of Crisis Communication* (eds W.T. Coombs and S.J. Holladay). https://doi.org/10.1002/9781119678953.ch1

van der Meer T.G.L.A., Verhoeven Joost W. M. (2014), Emotional crisis communication, *Public Relations Review*, Volume **40,** Issue 3, Pages 526-536,

Wang, Y., Song, W., Tao, W., Liotta, A., Yang, D., Li, X., Gao, S., Sun, Y., Ge, W., Zhang, W., Zhang, W. (2022). A Systematic Review on Affective Computing: Emotion Models, Databases, and Recent Advances. *Inf. Fusion*, 83-84, 19-52.

van Atteveldt W., Peng T., (2018) When Communication Meets Computation: Opportunities, Challenges, and Pitfalls in Computational Communication Science, *Communication Methods and Measures*, 12:2-3, 81-92, DOI: 10.1080/19312458.2018.1458084

Westermann, A., Forthmann, J. (2021), "Social listening: a potential game changer in reputation management How big data analysis can contribute to understanding stakeholders' views on organisations', *Corporate Communications: An International Journal*, Vol. **26** No. 1, pp. 2-22. https://doi.org/10.1108/CCIJ-01-2020-0028

Ward, J., Richardson, D., Orgs, G., Hunter, K., Hamilton, A. (2018). Sensing interpersonal synchrony between actors and autistic children in theatre using wrist-worn accelerometers., ISWC '18. Proceedings of the 2018 ACM International Symposium on Wearable Computers, 148–155, https://doi.org/10.1145/3267242.3267263

Wu, C.H., Huang, Y.M , Hwang, J.P. (2015). Review of affective computing in education/learning: Trends and challenges. *British Journal of Educational Technology*. **47**. 10.1111/bjet.12324.

Zimmermann, P., Guttormsen, S., Danuser, B., Gomez, P., (2003). Affective computing—a rationale for measuring mood with mouse and keyboard. *Int. J. Occup. Saf. Ergo.* **9** (4), 539–551.

Zhang, J., Yin, Z., Cheng, P., Nichele, S. (2020). Emotion Recognition Using Multimodal Data and Machine Learning Techniques: A Tutorial and Review. *Information Fusion*. **59**. 10.1016/j.inffus.2020.01.011.

Received August 1, 2023, revised July 23, 2024, accepted December 5, 2024

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 32-66
https://doi.org/10.22364/bjmc.2025.13.1.03

Towards Universal Modeling Language for Neural Networks

Janis BARZDINS, Audris KALNINS, Paulis BARZDINS

Institute of Mathematics and Computer Science,University of Latvia,
Raiņa bulvaris 29, Riga, LV-1459, Latvia

{janis.barzdins, audris.kalnins, paulis.barzdins}@lumii.lv

ORCID 0009-0008-0040-5557, ORCID 0000-0003-0907-7496, ORCID 0009-0006-7186-9776

Abstract. Effective modeling is essential in both system and software development, serving as a key method for facilitating understanding, guiding design, and enabling communication among stakeholders. However, traditional universal system modeling languages like UML and SysML fall short when it comes to neural network modeling, where the structure, training, and deployment processes demand more detailed and specialized representations. Conversely, domain-specific languages like Keras, TensorFlow, PyTorch, and tools like Netron and Deep Learning Studio are too closely tied to specific implementation environments. This creates a significant challenge: the need to develop a universal modeling language specifically for neural networks that is both sufficiently simple (requiring a description of around ten pages) and capable of providing a detailed description of neural networks and their management. The main contribution of this paper is the introduction of such a language, called UM1NN, along with a detailed description and its application demonstrated through two important use cases: describing GPT-2 and defining the fine-tuning of GPT-2 for Question-Answering.

Keywords: Neural Networks, ML Systems, Graphical Modeling, UM1NN language.

1. Introduction

As is well known, modeling plays a critical role in both system and software development, serving as a foundational technique for understanding, design, and stakeholder communication. A good overview of the current state in this field can be found in the paper (Michael et al., 2024) with the noteworthy title "Quo Vadis Modeling?".

The core of modeling is modeling languages. In general, modeling languages can be broadly classified into universal modeling languages and domain-specific modeling languages. Universal modeling languages are designed to provide a high-level, abstract view of various systems, facilitating stakeholder communication and overall system design. In contrast, domain-specific modeling languages (DSLs) offer detailed and precise tools tailored to the specific needs of particular domains, such as neural networks, ensuring effective implementation and management.

Over the past few decades, several universal system modeling languages have emerged, including UML (OMG, 2017; Rumbaugh et al., 2005), BPMN (Shapiro et al.,

2012; WEB (a)), SysML (Friedenthal et al., 2014; WEB (b)), KerML (OMG, 2024; Pires et al., 2024), conceptual modeling (Lukyanenko et al., 2024), and workflows (Aalst and Hee, 2022). These languages are primarily oriented towards business system modeling. At present, there are no widely recognized direct applications of these languages for neural network modeling. A more detailed discussion of the current state of neural network modeling will be provided in Section 2.

The problem addressed in this paper is the development of a universal modeling language specifically for neural networks that, on the one hand, is sufficiently simple (requiring a description of around ten pages, not the hundreds typically needed for languages like UML) and, on the other hand, allows for the detailed description of neural networks and their management. This language should be understandable to a typical stakeholder and usable as a communication tool among stakeholders.

The main contribution of this paper is the proposal of such a language, named UM1NN (with "1" signifying the initial version of the language). Section 3 provides a detailed description of UM1NN, while Sections 4 and 5 demonstrate its application through two significant use cases: describing GPT-2 (Section 4) and defining the fine-tuning of GPT-2 for Question-Answering (Section 5).

2. Overview of Neural Network Modeling Languages and UM1NN's Place

2.1. Universal Modeling Languages

The traditional universal system modeling languages mentioned in the Introduction are not sufficiently effective for neural network modeling. While these languages provide a variety of diagrams and notations to capture different aspects of system design, they are not optimized for the unique requirements of neural network modeling. Neural networks demand specialized and detailed representations, particularly for defining their structure, training processes, and deployment, which are not adequately supported by these general-purpose modeling languages.

2.1.1. Stereotypes and profiles

One way to make these universal modeling languages (primarily UML) more suitable for detailed neural network modeling is through the use of stereotypes and profiles. Stereotypes allow the customization of UML elements to represent domain-specific concepts. For instance, defining stereotypes like <<ConvolutionLayer>> or <<DenseLayer>>. Profiles are a collection of stereotypes and tagged values that cohesively extend UML for a specific domain. However, the introduction of stereotypes and profiles creates additional challenges:

- Complexity and Maintenance**:** Defining and maintaining stereotypes and profiles can become cumbersome, especially as the complexity of neural network architectures grows.
- Limited Expressiveness**:** Stereotypes and profiles may not capture all the nuances and specific details needed for precise neural network modeling.

- Performance**:** Using stereotypes for detailed neural network modeling can lead to performance issues in modeling tools.

Given these challenges, there is a need for a "hard extension" of UML to effectively model neural networks. A hard extension involves creating a new modeling language or significantly extending an existing one to directly support the requirements of neural network design, training, and deployment.

2.1.2. Our Proposed Solution: UM1NN

Our proposed modeling language, UM1NN, represents a hard extension of UML, specifically UML Activity diagrams, and is tailored for neural network modeling. UM1NN is designed to meet the specific demands of neural networks, providing both simplicity and comprehensive expressiveness:

- **Simplicity**: UM1NN is designed to be easy to understand and use, with a concise syntax and a limited set of core concepts. The entire language specification can be comprehensively described in about ten pages (see Section 3), and the description of a neural network in this language is almost self-explanatory (see Sections 4 and 5).
- **Effective Modeling:** UM1NN enables clear, stakeholder-friendly descriptions of neural networks, covering their architecture, training processes, hyperparameters, and deployment strategies, without aiming for formal specifications.
- **Enhanced Communication:** UM1NN provides a shared language for stakeholders, balancing the need for technical accuracy with accessibility, making it easy for both experts and non-experts to understand and communicate effectively.

The proposed modeling language UM1NN is, in some sense, domain-specific, but it differs from existing DSLs for neural networks in several key points:

- **Versatility**: It is not tied to a specific type of neural network. UM1NN can be used for all types of neural networks.
- **Framework Independence**: It is not tied to any particular neural network implementation framework. UM1NN specifies the operation of neural networks at a higher level of abstraction (similar to how UML is used for business systems).

According to the authors, this level of abstraction for describing neural networks holds an important place, much like UML descriptions for business systems, enabling stakeholders to communicate effectively without diving into the technical details of programming implementations.

To provide further clarity, the following sections will explore traditional domain-specific languages (DSLs) for neural networks and their practical applications.

2.2. DSLs for Direct Execution in Neural Network Frameworks

In this section, we focus on domain-specific languages (DSLs) designed to define, build, and train neural networks directly within popular machine learning frameworks. These languages are primarily intended for coding and execution rather than abstract system

design, offering developers the tools needed to specify and run models in real-time environments.

2.2.1. Program Libraries for Deep Learning Models

When discussing domain-specific languages for neural networks, it's essential to first mention the program libraries available in Python for building deep learning models at the code level. Several well-established libraries, such as **Keras (**Chollet and Watson, 2024; WEB(c)), **TensorFlow** (Chollet and Watson, 2024; Abadi et al., 2016), and **PyTorch (**Chollet and Watson, 2024, Paszke et al, 2019), provide high-level operations for defining and executing deep learning models directly within Python. These libraries include built-in support for typical deep learning model elements such as layers (e.g., Linear layers, Convolutional layers, Transformer layers) with customizable parameters like dimensions, Weight, and Bias tensors.

Additionally, these libraries simplify the process of training and processing models by offering features like dynamic computation graphs and tools for handling complex data flows. They allow the user to create executable models efficiently, abstracting away much of the boilerplate code associated with neural network training and evaluation.

However, these libraries focus primarily on building and executing models at the code level. They do not provide tools for visualizing the overall structure of the model at a higher abstraction level, such as graphical diagrams that might be used to communicate a model's architecture more clearly to non-technical stakeholders or during the design phase.

2.2.2. Graphical Modeling Tools for Deep Learning

An alternative to program libraries are tools that use graphical modeling languages to represent deep learning models. Two notable tools in this area are **Netron** and **Deep Learning Studio (DLS)**. Both tools utilize a graphical workflow language that allows users to define models visually, using a straightforward sequence of actions, combined with facilities for specifying parameters.

2.2.3. Netron Tool

Netron (WEB (d)) is an open-source visualization tool for deep learning models. It supports a wide variety of formats, allowing models developed in environments (Chollet and Watson, 2024) like Keras, TensorFlow, PyTorch, Caffe, and MXNet to be visualized in a standardized format. The model's architecture is represented as a series of rounded rectangles (representing actions, which correspond to layers), with arrows indicating the flow of data between them. Each layer's detailed information—such as input/output tensor shapes, parameters, and operations—is displayed, making it easy to inspect the model’s components.

Netron is primarily used to review and verify the structure of existing neural network models, providing insights to both technical and non-technical users. However, it does not offer features for building new neural network models from scratch. Its main focus is on understanding and validating the architecture of pre-trained models.

2.2.4. Deep Learning Studio (DLS)

Deep Learning Studio (DLS) (WEB (e,f)) is a framework designed for building deep learning models using a graphical interface. DLS provides a drag-and-drop editor for constructing neural networks, represented as workflow diagrams. In this environment, layers of the model are visualized as rounded rectangles (actions), connected by lines that represent the flow of tensors between layers. The shapes of the tensors being passed between layers are also displayed, offering a visual insight into the model's data flow. The editor allows users to select basic layers common to deep learning models, such as convolutional layers, pooling layers, and normalization layers. However, the ability to construct more complex architectures is somewhat limited compared to full-featured coding libraries like TensorFlow or PyTorch.

2.3. DSLs for Model-Driven Engineering (MDE) of Machine Learning-Enabled Systems

While the DSLs discussed in the previous section focus on directly executing machine learning models, Model-Driven Engineering (MDE) (Brambilla et al., 2012) offers a more abstract and systematic approach to system development. MDE-based DSLs describe system architectures, which are then transformed into executable models through automated or semi-automated model transformations. This section reviews the use of domain-specific languages (DSLs) within MDE for Machine Learning-Enabled Systems. These systems typically encompass not only neural networks but also workflows for data preprocessing, feature engineering, and broader software system integration. DSLs in this context enable automated code generation through model transformations, facilitating the development and deployment of comprehensive machine learning pipelines. A comprehensive overview of the current state in this field is provided by papers from Rädler et al. (2024) and Naveed et al. (2024). Below, we highlight key aspects of the current situation in this area.

2.3.1. Ecore-Based DSLs for Machine Learning

The Ecore metamodel from the Eclipse Modeling Framework (Steinberg D., 2009) is a widely used foundation for defining DSLs within MDE environments, offering high-level abstractions for the design and integration of machine learning models. Several tools extend EMF to support machine learning workflows:

- **EMF-IncQuery** (Horvath et al., 2015)**:** This tool facilitates pattern-based queries within models, enabling advanced querying capabilities that simplify the manipulation and analysis of machine learning workflows.
- **Text-based DSLs** (Friese et al., 2008)**:** Custom languages can be developed to specify machine learning models and workflows, providing flexible, domain-specific approaches that help automate processes such as data preprocessing and model training.

2.3.2. MontiAnna: A DSL for Deep Learning Model Specification

MontiAnna (Kusmenko et al., 2019; Kirchhof et al., 2022) is a domain-specific language specifically designed for deep learning. Built on the MontiCore framework, an MDE platform, MontiAnna allows the specification of neural networks and deep learning models. It focuses on integrating machine learning into broader system architectures. MontiAnna offers both graphical and textual notations, providing flexibility for defining neural networks and their associated training processes. Additionally, the framework supports model transformations, enabling the generation of executable code from the defined models. Essentially, MontiAnna bridges the gap between high-level system models and the practical implementation of machine learning components.

2.3.3. AutoML: A DSL for Automating the Deep Learning Lifecycle

AutoML (Moin et al., 2022) presents a novel approach to automating the entire machine learning lifecycle, from model selection to optimization and deployment. The system automates the selection of the most appropriate model architecture (e.g., Fully-Connected Neural Networks (FCNN) or Long Short-Term Memories (LSTM)) using Bayesian Optimization. AutoML simplifies the development of AI-intensive systems by automating model and hyperparameter selection, significantly reducing the need for manual tuning. This DSL is built on the ML-Quadrat framework and includes a user-friendly, web-based interface, making it a valuable tool for streamlining ML development.

2.3.4. Key Conclusions

Traditionally, MDE-based DSLs are formal languages that, at a given level of abstraction, can generate executable models through automated or semi-automated transformations. However, the proposed modeling language, UM1NN, is semi-formal. This raises the question: how does UM1NN relate to MDE-based DSLs? One potential research direction (outlined in the Conclusion section) is the exploration of methods to generate executable models from system descriptions written in semi-formal languages, leveraging Large Language Models (LLMs) such as ChatGPT. In this sense, we see a clear connection between UM1NN and traditional MDE approaches for Machine Learning-Enabled Systems.

3. Description of the UM1NN Modeling Language

UM1NN uses only one type of diagram, called UM1NN Activity Diagrams. The main concept in UM1NN, the **UM1NN System Model**, is defined as a set of thematically related UM1NN Activity Diagrams. An Activity Diagram represents a behavior composed of individual elements called actions. An action represents a single step within an activity. Therefore, defining the UM1NN language entails defining **UM1NN Activity Diagrams** (hereafter referred to simply as **UM1NN diagrams**).

3.1. UM1NNcore

As previously mentioned, in defining UM1NN, we will use UML Activity Diagrams as a foundation. The purpose of this section is to explain exactly what we are adopting from UML Activity Diagrams. Figure 1 shows a typical UML Activity Diagram taken from the official UML documentation (OMG, 2017). This diagram includes all the elements that we (with slight modifications) will directly incorporate into our modeling language UM1NN. (The official UML AD contains many other elements, but we will not use those.) However, this incorporation will come with one change: instead of Object Nodes, we will introduce a new type of node, which we will denote similarly but call a Data Node. An Object Node, according to UML AD semantics, is essentially a container that can hold many objects until they are processed by a subsequent action. In contrast, a Data Node (or Data Element) is analogous to a variable in programming languages, which can hold only one value at any given time, and this value can be used by multiple actions. To specify precisely which actions can modify and which can use the value of a data element, we will introduce dashed arrows as shown in Figure 2. We will retain the UML AD control flow arrows in our modeling language, but they will serve only to represent control flow (Fig.3, as an equivalent notation of Fig.2, we consider only as 'syntactic sugar'). Figure 2 also shows that action symbols can have a "Short Description" section in addition to their name, and data symbols can have a "Type Definition" section along with the name and "Short Description."

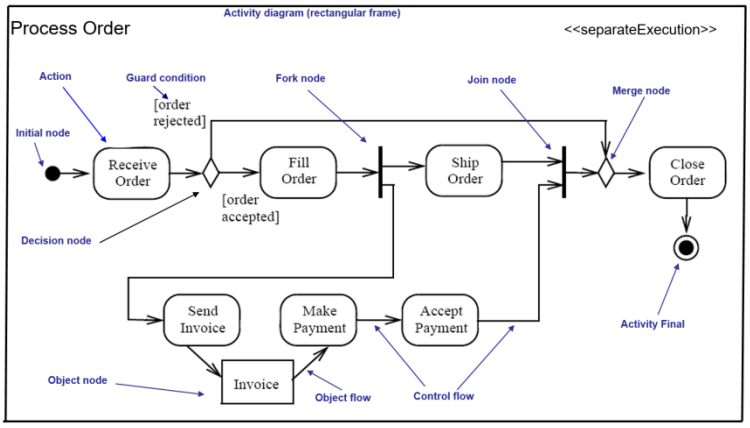


Figure 1. UML Activity Diagram.

To make UM1NN diagram notation more concise, we will agree on several defaults. First, using UML AD terminology, we will employ the <<separate execution>> semantics, meaning that a separate execution of the activity is created for each invocation. Second, when multiple control arrows converge into an action symbol, we will assume a merge symbol by default (and thus will not draw this symbol). Third, we will also omit the decision symbol (diamond); instead, we will agree that if an action symbol has more than one outgoing control arrow, each of these arrows must have an attached guard condition.

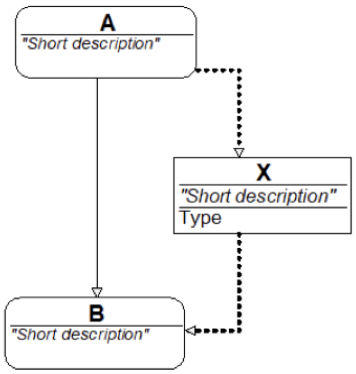


Figure 2. Dashed arrows.

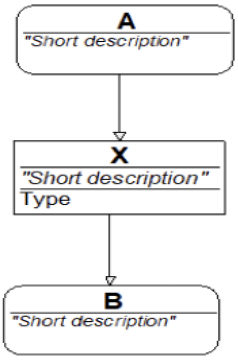


Figure 3. Equivalent representation of Fig. 2.

The resulting modeling language will be called UM1NNcore, and it will contain the following graphical elements:

- Activity diagram symbol (optional)
- Action symbol
- Data element symbol
- Control flow arrow (solid arrow)
- Control flow arrow with guard condition
- Data flow arrow (dashed arrow)
- Fork and Join symbols
- Start and End symbols
- (Decision and Merge symbols by default)

The UML Activity diagram shown in Figure 1 can be equivalently represented as an UM1NNcore diagram, as shown in Figure 4.

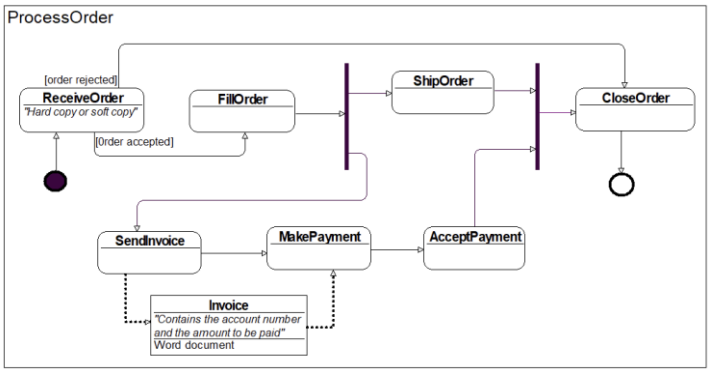


Figure 4. UM1NN Activity Diagram.

The following sections will focus on significantly extending UM1NNcore with new features tailored for neural network modeling applications, which could not be directly "borrowed" from UML AD.

3.2. Data Elements

In UM1NN Activity Diagrams, data elements play a crucial role in representing the information that flows through and is manipulated by the neural network. In UM1NN, data elements are classified into two main categories: **Flow Data** and **Internal Parameters**, providing a structured approach to represent the different types of information in the neural network.

3.2.1. Flow Data

Flow data in neural networks represent dynamic information traveling through the network (e.g., input data and intermediate results). Flow data elements (see elements X and Y in Figure 5) are represented by rectangular frames with **black borders** and **white background** and can appear multiple times within a single UM1NN Activity Diagram. Each occurrence of a flow data element with the same name operates independently. This independence allows for flexibility, as actions in the diagram can use different data elements with the same name without causing conflicts.

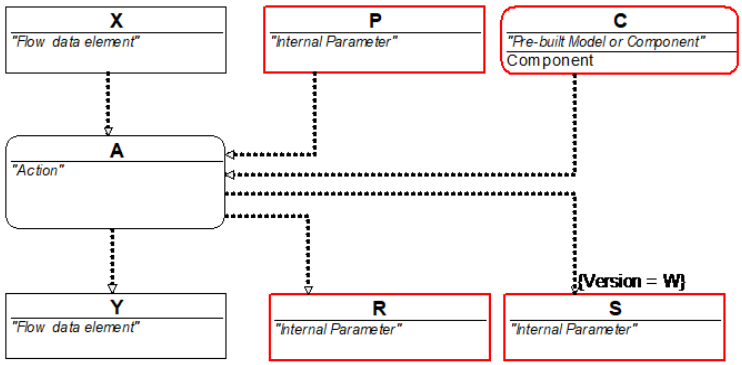


Figure 5. Action's dashed arrows.

3.2.2. Internal Parameters

Internal parameters are essential to the internal workings of the neural network. These include components like weight matrices and biases, which are adjusted during training to optimize the network's performance. In UM1NN diagrams, internal parameters (see elements P, R, S in Figure 5) are represented as rectangles with **bold red borders** to distinguish them from flow data elements. Internal parameters differ from flow data elements in that they maintain a consistent value across all diagrams in the UM1NN System Model. This means that if multiple internal parameters share the same name, they are automatically synchronized—any change made to one element is propagated to

all others with the same name. This ensures value consistency throughout the system model. Additionally, internal parameters can also have version labels for their values.

Internal parameters are typically used to represent model parameters (e.g., weights and biases) and hyperparameters (e.g., learning rate, batch size) and are depicted with bold red borders and colored backgrounds: **orange for model parameters** and **green for hyperparameters** (see Use Cases figures).

By integrating these concepts, UM1NN provides a structured way to represent and differentiate between the different types of data that flow through and govern the behavior of neural networks.

3.3. Pre-built Components

UM1NN Activity Diagrams can also include nodes that represent **pre-built components**. These nodes are graphically depicted as **rounded rectangles with bold red borders** and labeled with type = "Component" (see element C in Figure 5). These elements can be used in two ways:

- **Action using the pre-built component**: This refers to actions that utilize the pre-built component. A **dashed arrow** connects the pre-built component node to the action that makes use of it (see element C in Figure 5).
- **Direct use**: The pre-built component can also be directly used as an action, represented as shown in Figure 6 (further details will be discussed in the next section).

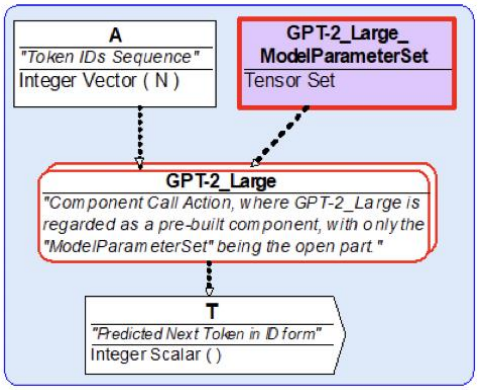


Figure 6. Component Call Action.

3.4. Actions

In UM1NN, **actions** represent individual steps or operations within an activity. These actions can be classified into four categories based on their function and scope: **Formal Actions**, **Informal Actions**, **Subdiagram Call Actions**, and **Component Call Actions**.

3.4.1. Formal Actions

Formal actions are well-defined operations with specific inputs and outputs. These actions typically involve mathematical operations essential to neural networks, such as tensor operations and matrix manipulations.

- **Examples**: Matrix multiplication, convolution, pooling, normalization.
- **Graphical Notation**: Formal actions are depicted as **rounded rectangles with a yellow background** (see Fig. 7).

3.4.2. Informal Actions

Informal actions represent higher-level processes that might encompass multiple steps or operations. Unlike formal actions, informal actions are more abstract, and their inputs and outputs may not be as clearly defined. These actions often describe broader processes in the neural network.

- **Examples**: Backpropagation, gradient descent, preparation of input data.
- **Graphical Notation**: Informal actions are represented as **rounded rectangles with a white background** (see Fig. 8).

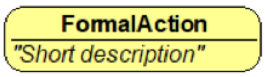


Figure 7. Formal Action.

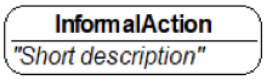


Figure 8. Informal Action.

The boundary between **Formal Actions** and **Informal Actions** can sometimes be fluid, depending on the stakeholder's level of expertise. For example, an action like backpropagation might be considered "formal" to someone with deep knowledge of neural networks, but "informal" to someone less familiar with the process. One useful criterion to distinguish between these two types of actions is whether an advanced model like **ChatGPT-4** can fully understand the action based solely on its **name** and **short description** (recall that action symbols in UM1NN diagrams contain both a "Name" and a "Short Description" section). If the action can be explained in detail by ChatGPT-4 based on this information alone, it is typically considered **formal**. In contrast, if the action is more abstract and requires broader contextual understanding, it leans towards being **informal**.

3.4.3. Subdiagram Call Actions

Subdiagram call actions refer to another diagram that is logically part of the current diagram. These actions allow for modularity by breaking down complex processes into smaller, manageable diagrams. Input and output parameters are clearly indicated, ensuring the proper flow of data between the main diagram and the subdiagram.

- **Graphical Notation**: Subdiagram call actions are represented as **rounded rectangles with a double border** and display both **input** and **output parameters**, as well as the corresponding **internal parameters** (see Fig. 9).

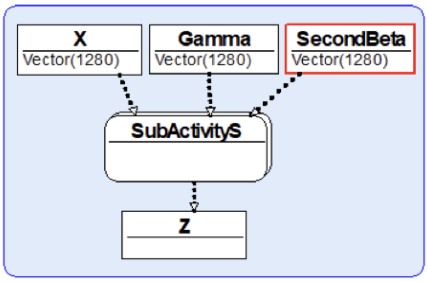


Figure 9. Subdiagram Call Action.

3.4.4. Component Call Actions

Component call actions represent the use of pre-built, sophisticated models or components. These actions are abstract in nature, meaning the internal workings of the component are hidden from the diagram, focusing instead on well-defined inputs, outputs, and exposed internal parameters. Component call actions are often used when applying pre-trained models or neural network architectures.

- **Examples**: Fine-tuning a pre-trained GPT-2 model, using ResNet for feature extraction, employing BERT for text encoding.
- **Graphical Notation**: Component call actions are depicted as **rounded rectangles with bold red double borders**. They also include **input**, **output**, and any exposed **internal parameters** (see Fig. 6).

3.4.5. One More Remark

Consider **Figure 5**, which illustrates a possible usage of dashed arrows in relation to **Action A**:

- **X ---> A** means that A uses the value of the flow element X.
- **A ---> Y** means that A modifies the value of the flow element Y.
- **P ---> A** means that A uses the value of the parameter P.
- **C ---> A** means that action A utilizes the component C.
- **A ---> R** means that A modifies the value of the parameter R.
- **A --{version=W}--> S** means that A modifies the value of the parameter S and assigns it the version label "W" (recall that internal parameter values can also be assigned version names).

This example demonstrates how UM1NN diagrams visually depict interactions between actions, flow elements, parameters, and components, ensuring clear tracking of value modifications and usage.

3.5. Dashed and Solid Arrows Between Flow Data Elements

Let us introduce a new type of operational mechanism between flow data elements (nodes) – the so-called dashed arrow mechanism (Fig. 10). If there is a dashed arrow

from data node X to another data node Y of the same type, it means that if any action at any point in time changes the value of node X (remember, X is formally a variable), this new value is immediately transferred to node Y. This mechanism allows us to freely supplement the existing activity diagram with new identical data nodes, but with different names. This, in turn, gives us the ability to graphically define both the graphical call mechanism and the diagram detailing mechanism (see the next section).

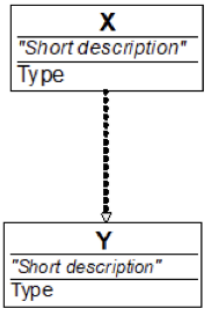


Figure 10. Dashed arrow.

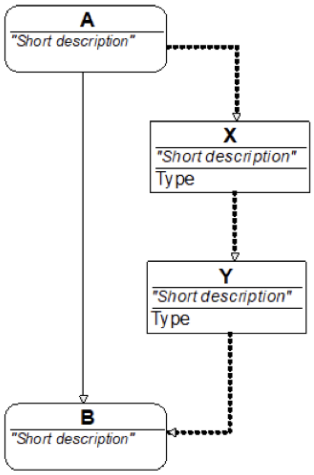


Figure 11. Diagram fragment.

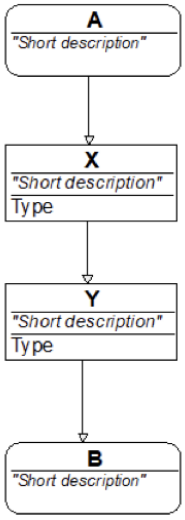


Figure 12. Equivalent representation of Fig. 11.

In many cases, it is natural to combine the control flow mechanism (control arrows, which are solid arrows) with the data flow mechanism (dashed arrows). Let's assume we have a fragment of an activity diagram corresponding to Fig. 2. Let's agree that it can equivalently be depicted as in Fig. 3. In this drawing, it is seen that the control arrow simultaneously serves as a data transfer arrow. At this level, it is just "syntactic sugar." However, let's add additional semantics to this construction: we will consider that the control token does not go "directly" to action B, but first to data node X, and from there further to action B. Let's combine this mechanism with the previously mentioned "dashed arrow" mechanism. As a result, the diagram fragment shown in Fig. 11 can naturally be depicted as in Fig. 12. In this case, the control token from action A first goes to data element X, then from data element X to data element Y, and finally from data element Y to action B. Now let's note that the control arrow from data element X to data element Y simultaneously serves the role of the "dashed arrow" – it instantly transfers the value of element X to element Y. As a result, our language will have two types of arrows between flow data elements: dashed and solid. Both types of arrows ensure the transfer of element values, but the solid arrow additionally ensures the transfer of control tokens.

Additionally, we will allow dashed arrows from internal parameters to flow data elements, with the same semantics as in the case of flow elements.

We will use these constructions in the next section for fragmentation call mechanism.

3.6. Graphical Call

First, let us note that a UM1NN System Model typically contains a single diagram called the **Main Diagram**, which serves as the central point of the model. This diagram, whose name starts with the word "Main," uses the call mechanism to invoke other diagrams within the model, known as **subdiagrams**, or predefined components.

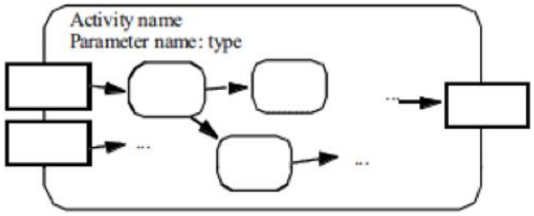


Figure 13. UML AD with parameters.

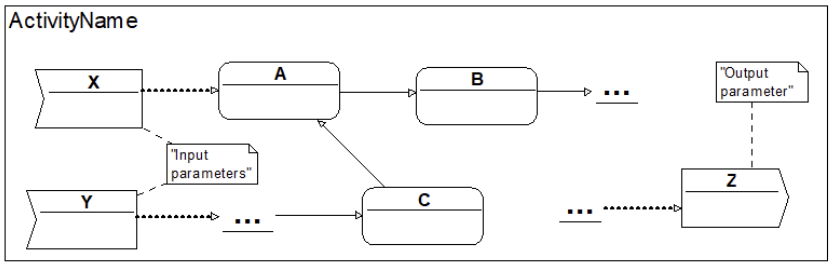


Figure 14. UM1NN AD with input and output parameters.

In this section, we will examine the UM1NN subdiagram call mechanism. First, let's introduce the concept of an "Activity Diagram with Input and Output Parameters" (so that we have a use for the call mechanism). In UML AD, parameters are represented as shown in Fig. 13. However, this representation and its usage in graphical call notation present several inconveniences (in practice, we often need to switch from graphical to textual form). In the UM1NN language, we will take a different approach. First, we will introduce a new notation for input and output parameters: they will be depicted as flow data nodes (anywhere in the diagram), with slightly modified graphical symbols (**InArrows**, **OutArrows**) as shown in Fig. 14 (input parameters X and Y, output parameter Z). In fact, these symbols are introduced only for better visual clarity, as we will consistently follow one rule regarding parameters: input parameter nodes should not have incoming arrows, and output parameter nodes should not have outgoing arrows. Therefore, in the graphical notation of a call action (see Fig. 15), specific symbols for input and output nodes can be replaced with the standard data node symbols.

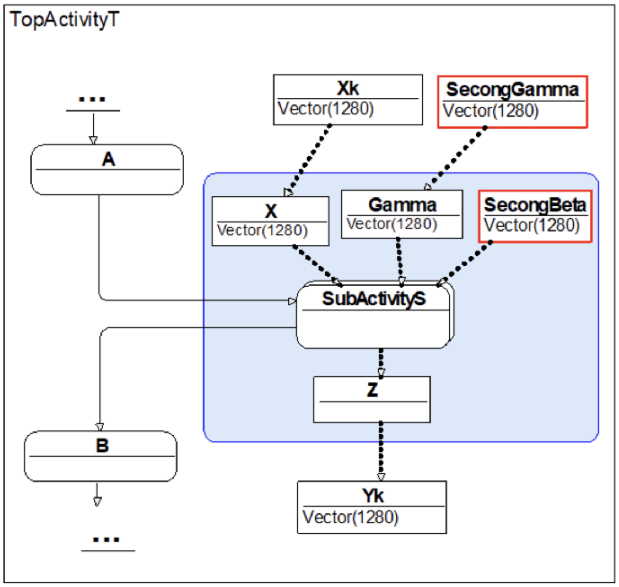


Figure 15. Subdiagram Call Action in the context of a Parent Diagram.

Now the most important part: the values of input parameters at the call action can be defined with the help of incoming dashed *arrows* from the corresponding data elements (nodes) that represent these values. Similarly, for output parameters, the dashed arrows will be outgoing (see Fig.15). In this notation, a call is represented not by the "fork" symbol as in UML AD, but as an action symbol with a double frame, from which dashed arrows indicate the output parameters of the called activity, and incoming dashed arrows indicate the input parameters. The name of the action with the double frame must match the name of the activity diagram we want to call. This activity diagram (which we want to call) must be from the same UM1NN system model in which this call occurs. We assume here that the activity (in the form of an action diagram) called through the call operation contains exactly one start symbol and exactly one end symbol. Under these conditions, the semantics of the call operation are clear.

Next, consider the case when the called diagram is only a fragment of a larger activity diagram and therefore might not contain start and end symbols directly. Recall the solid arrow mechanism introduced in the previous section, which serves both as a control flow and data element value transfer mechanism. This means that if we consider a large diagram W (Fig. 16) that logically contains a fragment S, we can depict this fragment S as a separate diagram (Fig. 17), where X can be considered as the start symbol and also as the input parameter symbol, and Y can be considered as the end symbol and also as the output parameter symbol. As a result, we can represent the original diagram W in a much more compact form using the new type of call operation (Fig. 18). This is a very convenient mechanism to logically divide large diagrams into manageable fragments (widely used in the section on GPT).

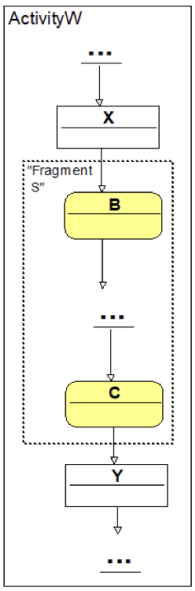


Figure 16.
"Large" diagram.

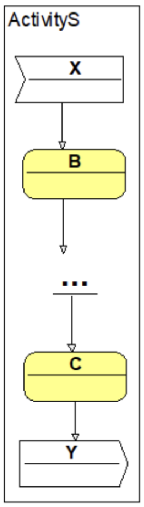


Figure 17.
Fragment as separate diagram.

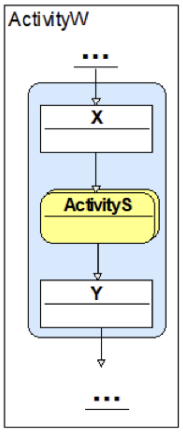


Figure 18.
"Large" diagram in more compact form.

3.7. Data element types and operations

For data elements, end-users can define any specific data types and their corresponding operations. However, it is up to the end-user to explain these to potential stakeholders. For applications in the field of neural networks, our language UM1NN will introduce the most crucial data type at its core - **tensors.**

More precisely, UM1NN proposes the following syntax for defining data element types, referred to as tensor types:

<dtype> Tensor <shape> (with "Float" as the default <dtype>)

If we specify a particular <dtype> (for example, Float) and a specific shape (for example, (1024, 512)), we then define a specific data element type:

Float Tensor (1024, 512)

whose values will be matrices of Float elements with 1024 rows and 512 columns.

Other examples include:

- Tensor () - alternatively called Scalar
- Tensor (25) - alternatively called Vector (25) (semantics: a vector of length 25, with element indexing starting at 0)
- Tensor (3, 5) - alternatively referred to as Matrix (3, 5) or 3 x 5 (semantics: a matrix with 3 rows and 5 columns, with row and column indexing starting at 0)

- Tensor (2, 3, 5) - tensors with shape (2, 3, 5) (element indexing starts at 0 for any dimension).

We also use the notation simply

Vector (without specifying the shape) – this means a vector of any length.

In these examples, we used the default dtype = float. If, for instance, we wanted to consider vectors of length 25 with Integer type elements, we would write

Integer Vector (25). Similarly, Integer Scalar, Integer Matrix (3, 5), etc.

Regarding operations with tensor-type data elements, PyTorch offers about 100 operations. We will list only the main ones here, in a slightly simplified syntax (noting that there is no uniform official syntax for tensor operations).

Let c be a scalar of type float, u and v be vectors of equal length with the same element type, specifically float, and A and B be matrices with identical shape and element type, also float. Basic operations:

- **u + v** - This represents the vector addition of u and v. Each corresponding element of the vectors is added together.
- **u - v** - This is the vector subtraction of u from v. Each corresponding element of u is subtracted from the corresponding element in v.
- **A + B** - This indicates the matrix addition of A and B. Each corresponding element of the matrices is added together.
- **c * v** - This refers to the scalar multiplication of the vector v by the scalar c. Each element of v is multiplied by c.
- **c * A** - This represents the scalar multiplication of the matrix A by the scalar c. Each element of A is multiplied by c.
- **A + v** - Assuming the number of columns in A equals the length of the vector v, this expression would typically denote broadcasting the vector v across each row of A and then adding v to each row of A. In PyTorch, this is handled automatically if v is shaped correctly.
- **A * B** - This is the element-wise multiplication of A and B. Each element in A is multiplied by the corresponding element in B. This operation is known as Hadamard product.
- **concat(A, B)** - This function concatenates matrix B to the right of matrix A, forming a new matrix whose width is the sum of the widths of A and B, with the same number of rows. In PyTorch, this is performed using **torch.cat([A, B], dim=1)**. The **concat** operation can also be applied to three or more matrices, for example, **concat(A, B, C, ...)**.
- **A @ B** - If the number of columns in matrix A equals the number of rows in matrix B, then $A@B$ represents the traditional matrix multiplication of A and B (also called *dot product*). In PyTorch, this operation is denoted by the **@** operator or by using **torch.matmul(A, B)**.

- **v @ A** - If v is a vector of length N and A is a matrix of shape $N \times M$, then $v@A$ represents the matrix multiplication of the vector v with the matrix A (also called *dot product*). This operation results in a new vector of length M. In PyTorch, this can be performed using the @ operator or **torch.matmul(v, A).**
- **u.dot(v)**: This represents the dot product of vectors u and v. It is the sum of the products of the corresponding elements of the vectors. In PyTorch, this can be computed using **torch.dot(u, v)**.
- **A^T** - This denotes the transpose of the matrix A. In matrix A, rows and columns are swapped, and in PyTorch, this is simply accessed with **A.T.**

These operations, extensively used in Sections 4 and 5, are crucial in our modeling language. They can be selected as predefined by end-users from a broader list of operations, which must be accompanied by clear explanations of each operation for stakeholders in the model documentation.

3.8. Specific Data Type: Tensor Set

Let's start with an example of a Tensor Set type data element shown in Fig. 19. It represents the entire set of weight and bias tensors used in GPT-2_Large. Similarly, Fig. 20 depicts another Tensor Set type element - the GPT-2_Large Hyperparameter Set. In this case, it is not a set of arbitrary tensors but rather a set of scalar tensors.

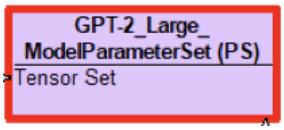


Figure 19. Tensor Set example.

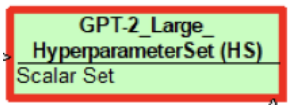


Figure 20. Another Tensor Set example.

These examples demonstrate the typical use of a Tensor Set for specifying specific collections of inner parameters in the UM1NN system model. Tensors have the concept of shape. Let's also introduce the concept of shape for a Tensor Set: it will be a set of pairs where each pair's first element is the tensor name, and the second element is the tensor's shape.

A tensor with a specific (fixed) shape has the concept of Value. Similarly, we can define the Value concept for a Tensor Set with a fixed shape as the values of the tensors it contains. Thus, when describing actions, we can use the term "assign values to a tensor set type element." For example, setting the "GPT-2_Large Model Parameter Set" involves assigning the pre-trained tensor values (weights and biases) to this set.

Let us also agree that in UM1NN diagrams, to indicate that a specific tensor (for example, "EmbeddingMatrix") belongs to a Tensor Set type element (for example, "GPT-2_Large Model Parameter Set"), we will use the same background color for both elements (in this case, orange).

3.9. Additional note on dashed arrows between data elements.

As mentioned in Section 3.5, if X and Y are nodes of the same type (for example, Integer Vector (1240)), a dashed arrow from node X to node Y means that any change in the value of node X is instantly transferred to node Y (i.e., X and Y, as variables, have the same value at any given moment). Since in Section 3.5 (see Fig. 10, Fig. 11, Fig. 12) we combined the dashed arrow with the control arrow (i.e., transformed it into a solid arrow), we will also apply the aforementioned automatic data transfer mechanism to solid arrows between data nodes.

For tensor-type data nodes, we will slightly extend this automatic value transfer mechanism: When a matrix A with m rows and n columns (i.e., a tensor A(m,n)) and a vector V of length n (i.e., a tensor $V(n)$) are connected by a dashed or solid arrow (represented graphically), the notation $V \rightarrow A(5,:)$ specifies that the values of vector V replace the values of the 5th row of matrix A. This action is symbolized by an arrow pointing from V to A, indicating the direction of data transfer. Similarly, the reverse operation is denoted as $(5,:) \rightarrow V$, where the 5th row of matrix A is used to update the values of vector V. See Fig. 21 and Fig. 22 for applications of these notations. This notation enables clear and concise representation of data flow between specific elements of matrices and vectors within the model.

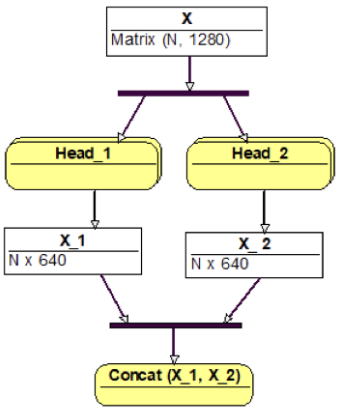


Figure 21. Concurrency example.

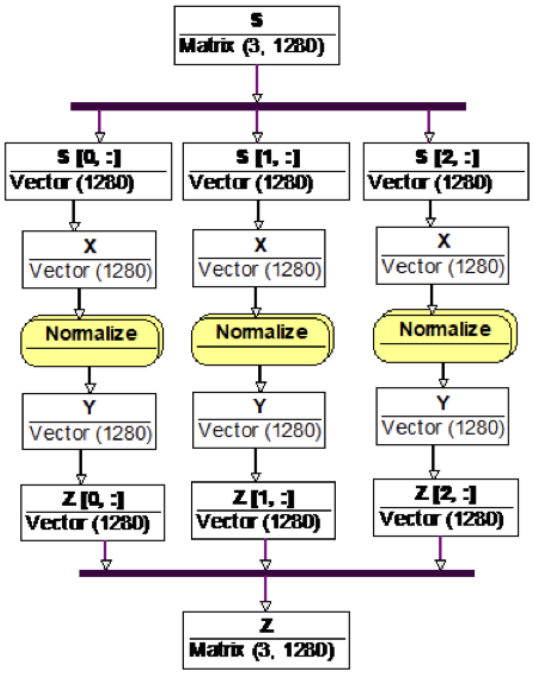


Figure 22. Another concurrency example.

3.10. Representing and Managing Concurrency

By **concurrent activities**, we mean activities that can execute independently and potentially in parallel with each other. In UML Activity Diagrams (AD), this is achieved through the use of fork and join nodes, which enable parallel execution and synchronization of these activities. In our language, we also adopt the fork-join mechanism, but with an additional requirement that each fork node corresponds to exactly one join node, and between them are independent concurrent activities (Fig. 21, Fig. 22, Fig. 23).

If the number of concurrent activities between the fork and join nodes is small, it is easy to represent them in our graphical modeling language. However, in neural network scenarios, it is typical to have a large number of concurrent activities, making it impractical to draw all of them directly. These activities often differ only slightly, specifically by the value of an integer parameter that appears in the definitions of actions and/or data elements. The aim of this section is to propose a special type of graphical loops (called **concurrency graphical loops**) for defining such uniform concurrent activities.

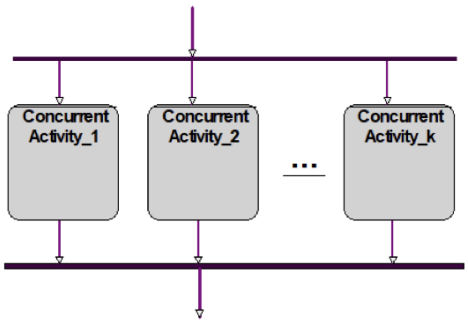


Figure 23. Concurrent activities.

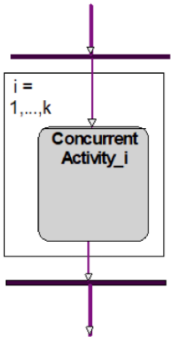


Figure 24. Concurrency loop.

First, let's agree that integer-type variable parameters will be denoted by lowercase Latin letters, for example, *i, j,* etc. Next, if we want to indicate that a letter, such as *i*, should be interpreted as a loop parameter in a syntactic loop, we will represent it in the form *_i<space>, _i<underscore>, _i+1<space>, or _i+1<underscore>*. The notation of Fig. 23 using this type of loop is shown in Fig. 24. If the range of the parameter *i* is small, the cyclic construction can be omitted. However, in neural network models, the range of the aforementioned parameter values is usually very large - several hundred or thousand, and the use of these loops is essential. The GPT model discussed in Section 4 relies heavily on the extensive use of concurrency graphical loops.

By the way, our concurrency graphical loops share similarities with UML graphical loops that use expansion regions (OMG, 2017: WEB (j)), but offer different capabilities.

Finally, one more clarification about data flow semantics in the case of fork and join nodes. First, we syntactically exclude the case where dashed arrows enter or exit fork or join nodes. However, it is permissible for solid arrows to enter or exit fork and join nodes, which can connect not only to action nodes but also to data nodes. In such cases, these arrows serve both control transfer and data transfer functions. Consider Fig. 21. In

this diagram, a solid arrow goes from data node X to the fork node, and from there to actions Head1 and Head2. Since this arrow originates from data node X, it means that it carries the value of node X with it. This, in turn, implies that actions Head1 and Head2 can directly use node X in defining their operations. A similar situation occurs with the join node—solid arrows "carry" the values of data elements X1 and X2 through this node. A more complex situation is shown in Fig. 22. Here, the row transfer mechanism for matrices mentioned in previous Section is used. Additionally, it demonstrates our agreement on flow data elements from Section 3.2—each new occurrence of a data element, even with the same name, is treated as a separate data element. Therefore, we repeat the data element named X three times (not X1, X2, X3), and similarly, Y three times (not Y1, Y2, Y3). This allows us to make the call mechanism significantly more versatile. For example, in this case, the Normalize activity is defined only for the pair (X, Y), but Normalize (X1, Y1), Normalize (X2, Y2), Normalize (X3, Y3) are obtained through the graphical parameter transfer mechanism (data and control transfer arrows S[0,:]->X, S[1,:]->X, S[2,:]->X).

Thus, the main constructions of the UM1NN language are outlined. The next section will cover a few more technical additional features.

3.11. Additional Features

In modeling languages, comments play an important role. In UM1NN, two types of comments and their respective notations are provided:

- **Element Comments**: Similar to UML, any diagram element can have an attached comment (rectangle with a folded corner), as shown in Fig.14.
- **Fragment Comments**: UM1NN also allows adding comments to an entire diagram fragment. This is done by drawing a dashed rectangle around the fragment and including explanatory text inside the rectangle using quotation marks (" "). This text typically serves as a meaningful name at a higher level of abstraction, explaining the defined content of the fragment (see, for example, Fig.16).

Additionally, neural network models often consist of many similar fragments. Therefore, in some cases, it is useful to use **ellipses** in the usual intuitive sense. In our proposed modeling language, such use of ellipses is allowed (see, for example, Fig. 16, 17, 18, 23), provided that the potential reader (stakeholders) can understand or infer their meaning in the given context.

3.12. Concluding Remark

The proposed neural network graphical modeling language provides a comprehensive framework for describing the detailed structure and operations of neural networks, including training processes and tensor-based data manipulations. In the following sections, we will utilize this modeling language to represent two significant use cases: the GPT model and the Fine-Tuning model for Question Answering based on GPT. To facilitate understanding, each section will be accompanied by a natural language explanation of the key concepts behind GPT-2 and Fine-Tuning, generated with the assistance of ChatGPT-4.

4. Use Case 1: GPT-2 Large Description in UM1NN Language

GPT-2_Large (Farris et al., 2024; Alammar, 2019) is a powerful language model designed to generate coherent and contextually relevant text from input data. Its functionality relies on multiple interconnected components, each handling distinct phases of the text generation process. This structure is represented in the UM1NN modeling language, spanning from Fig. GPT-1 to Fig. GPT-9, where each diagram illustrates a specific aspect of the model's workflow. Below is a brief overview of these models.

4.1. Main Model of GPT-2_Large (Fig. GPT-1)

The **Main Model** manages the overall context processing and data flow. It receives a sequence of token IDs as input and produces the predicted next token in ID form. Key stages include:

- **Embedding and Positioning**: Converts token IDs into embeddings and adds positional information to prepare the input for the Transformer.
- **Call Transformer Model**: Invokes the Transformer to generate a probability distribution over possible next tokens.
- **Select Predicted Token**: Chooses the token with the highest probability from the Transformer output.

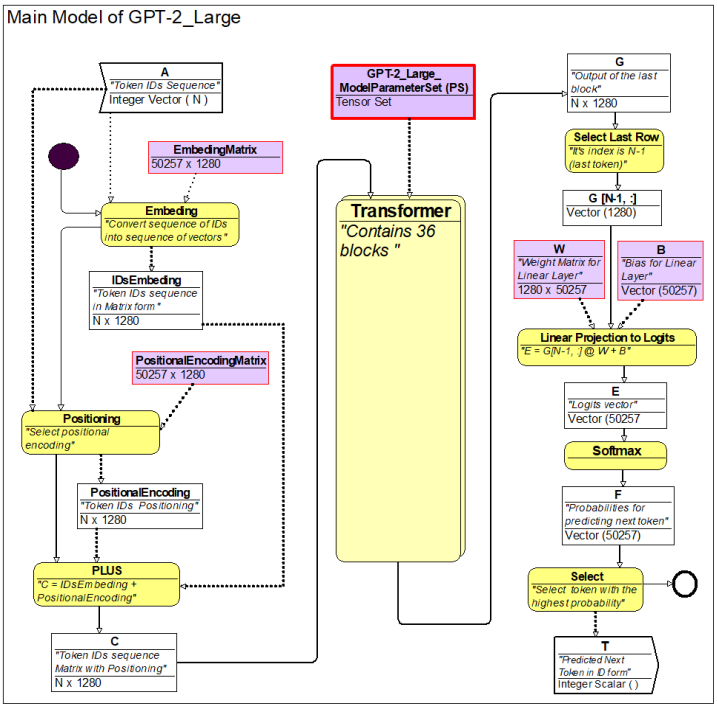


Figure GPT-1.

4.2. Transformer Model (Fig. GPT-2)

The **Transformer Model** (Turner, 2024; Alammar, 2019) is the core component, consisting of 36 sequential blocks. Each block performs:

- **Multi-Head Attention**: Allows the model to focus on different parts of the input.
- **First Add & Normalize**: Stabilizes the attention output for the next step.
- **Feed-Forward Processing**: Transforms the data further.
- **Second Add & Normalize**: Finalizes the block before passing data to the next one.

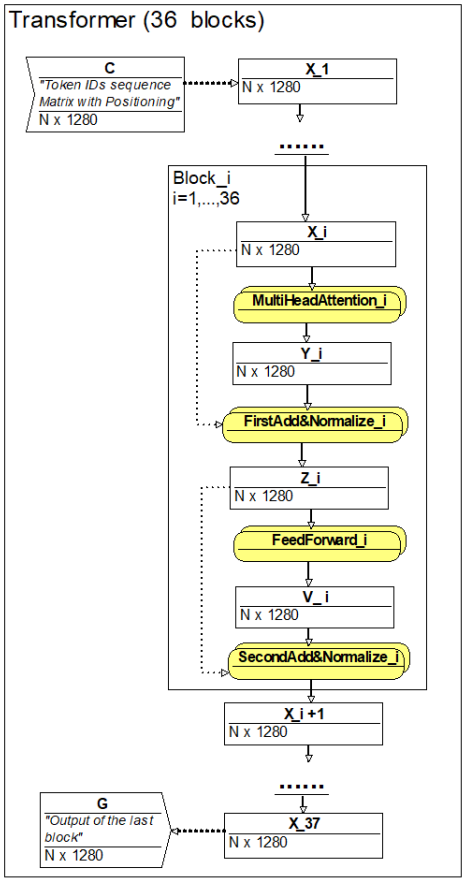


Figure GPT-2.

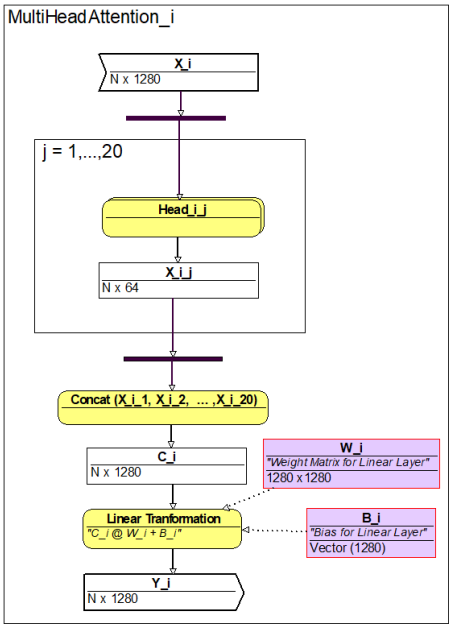


Figure GPT-3.

4.3. Multi-Head Attention Model (Fig. GPT-3)

This model handles the attention mechanism (Vaswany et al., 2023), enabling the Transformer to focus on multiple parts of the input sequence simultaneously. It includes:

- **Individual Head Models**: Each head processes a segment of the input data.

4.4. Individual Head Model (Fig. GPT-4)

Each attention head computes attention scores for a subset of the input sequence:

- **Calculate Head Output**: Produces the attention output for this head.

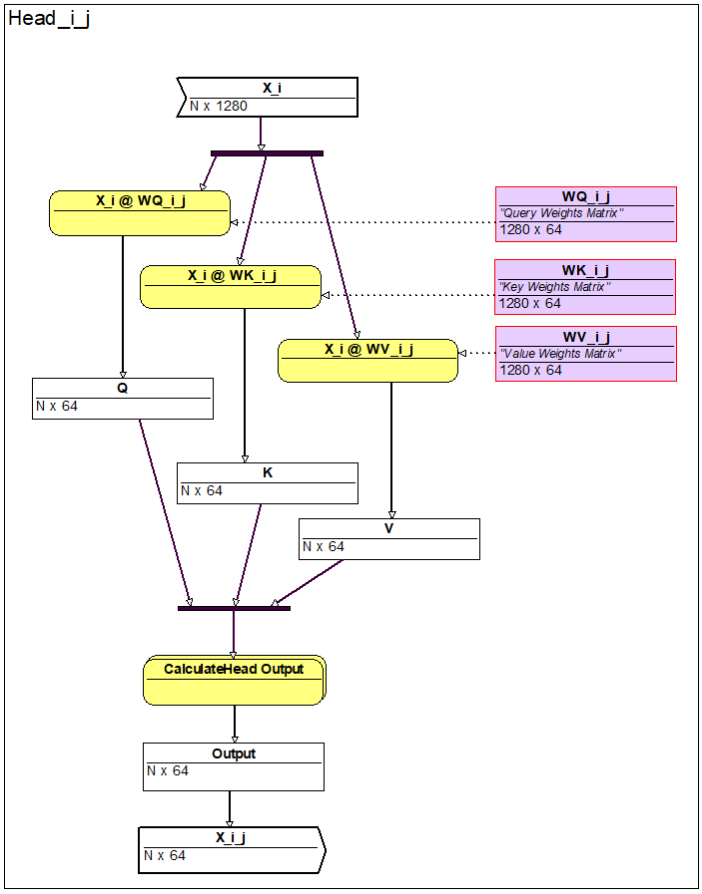


Figure GPT-4.

4.5. Calculate Head Output Model (Fig. GPT-5)

The core mechanism for attention is based on the Scaled Dot-Product Attention introduced by Vaswani et al. (2017; 2023) in their seminal paper *"Attention is All You Need."* It consists of the following steps:

- **Dot-Product Calculation**: Computes the dot product between the query and key matrices to determine the relevance scores for each query-key pair.
- **Softmax Normalization**: The attention scores are normalized across each row (for each query) using the softmax function, converting them into a probability distribution that represents the attention weights.
- **Output Generation**: The normalized attention scores are used to weight the value vectors. This weighted sum of values produces the output for the specific attention head.

The outputs from all heads are then aggregated to form a more comprehensive understanding of the input sequence.

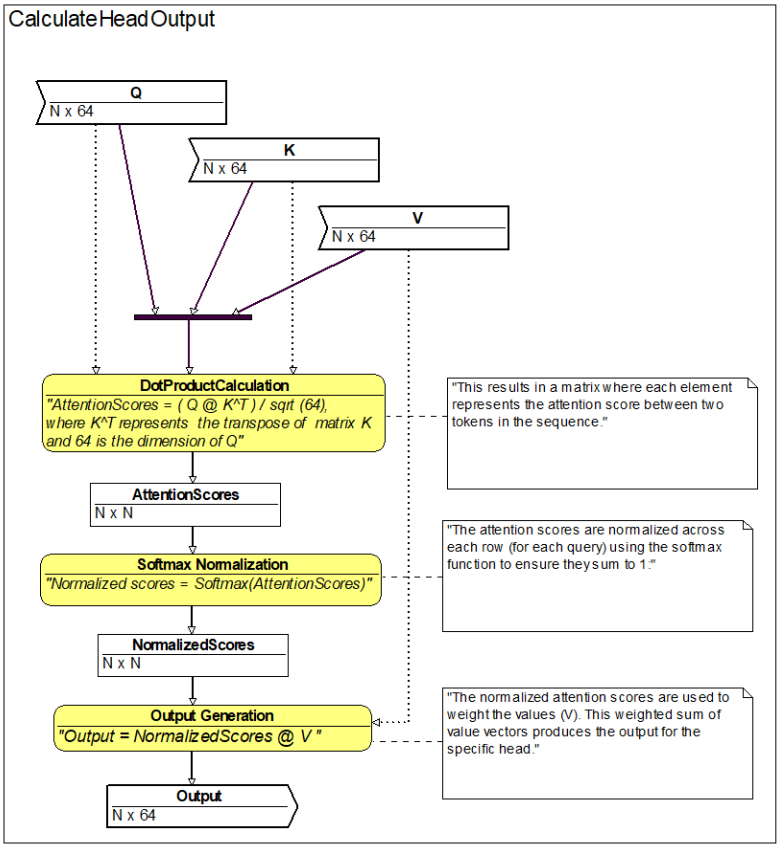


Figure GPT-5.

4.6. First Add & Normalize (Fig. GPT-6)

This model stabilizes the data after the attention step by:

- **Residual Connection**: Adds the original input to the attention output.
- **Normalization (Fig. GPT-9)**: Normalizes the result to improve stability before passing it to the feed-forward layer.

4.7. Feed-Forward Model (Fig. GPT-7)

A fully connected network that further processes the normalized data. It includes:

- **Linear Transformation**: Adds non-linearity with an activation function (GELU).
- **Second Linear Transformation**: Projects the data back to the original dimension, preparing it for the next step.

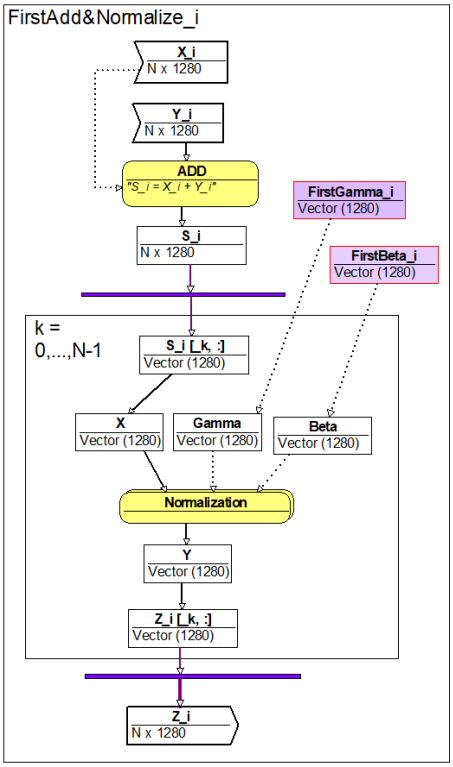


Figure GPT-6.

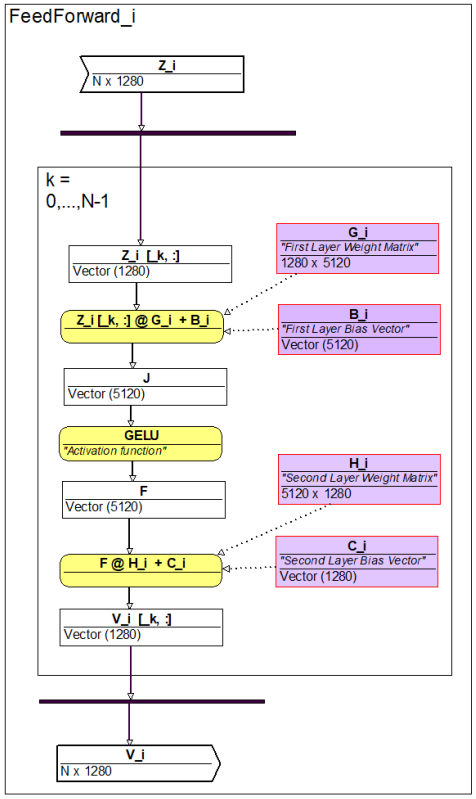


Figure GPT-7.

4.8. Second Add & Normalize (Fig. GPT-8)

Similar to the first, this model stabilizes the output of the feed-forward network. It includes:

- **Residual Connection**: Adds the original input to the feed-forward output to retain information.
- **Normalization (Fig. GPT-9)**: Ensures consistent scaling before passing the data to the next Transformer block or final output.

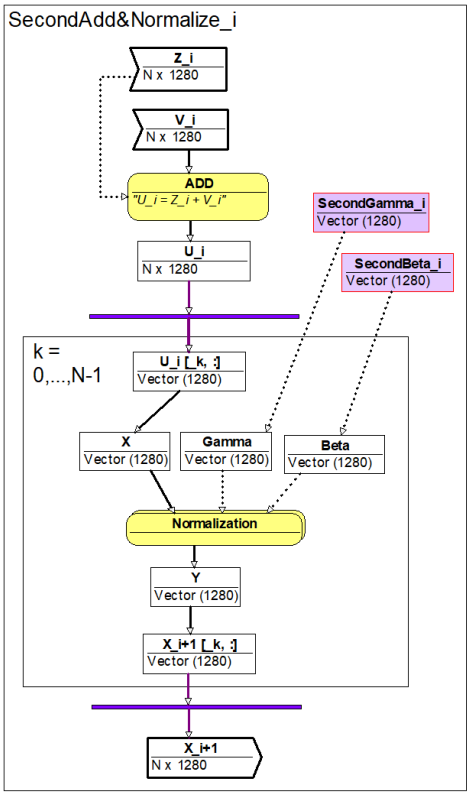


Figure GPT-8

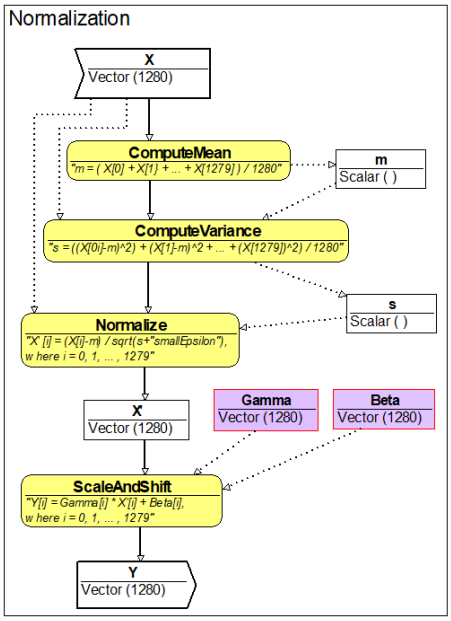


Figure GPT-9

5. Use_Case 2: Fine-Tuning GPT-2 for Question-Answering

Fine-tuning GPT-2 for question-answering tasks (Radford et al., 2019) involves adapting the pre-trained GPT-2 model to generate accurate answers for a specific dataset or task. This fine-tuning process includes several stages: loading the model, processing data in batches, calculating gradients, updating model parameters, and generating answers. This process is represented in the UM1NN modeling language in **Figures FT-1 through FT-4**, where each diagram (model) is responsible for a distinct phase of the workflow. Below is a brief explanation of these models in natural language.

5.1. Main Model of Fine-Tuning GPT-2 for Question-Answering (Fig. FT-1)

The **Main Model of Fine-Tuning GPT-2 for Question-Answering** governs the entire fine-tuning process. This high-level model manages the key stages, ensuring efficient data flow and invoking other submodels to handle specific tasks. It includes the following stages:

- **Load GPT-2 Model**: Initializes and loads the pre-trained GPT-2 model.
- **Collect Data**: Gathers task-specific question-answer data for fine-tuning.
- **Tokenize Data**: Converts the text data (questions and answers) into tokens that GPT-2 can process.
- **Batch Processing**: Invokes the **Batch Processing Model** to handle individual batches of tokenized data.

- **Evaluation**: Evaluates the model's performance using a validation dataset after each epoch to assess improvements or determine when to stop the fine-tuning process.

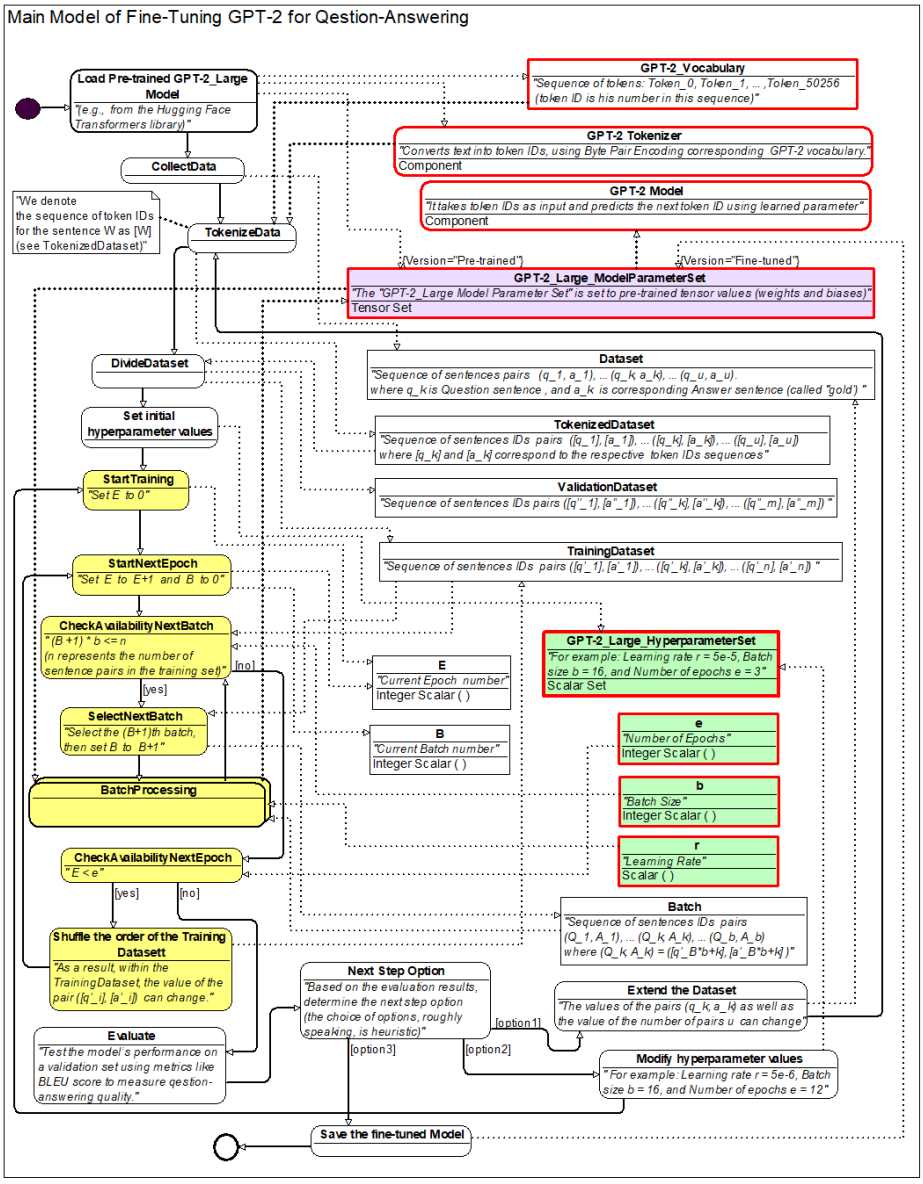


Figure FT-1

5.2. Batch Processing Model (Fig. FT-2)

The **Batch Processing Model** focuses on processing each batch of tokenized data. It includes the following key actions:

- **Process Current Batch**: Handles the batch of tokenized data.
- **Compute Gradients**: Invokes the **Gradient Calculation Model** to compute gradients for the current batch.
- **Parameter Update**: Updates the GPT-2 model's weights and biases based on the gradients received from the **Gradient Calculation Model**.

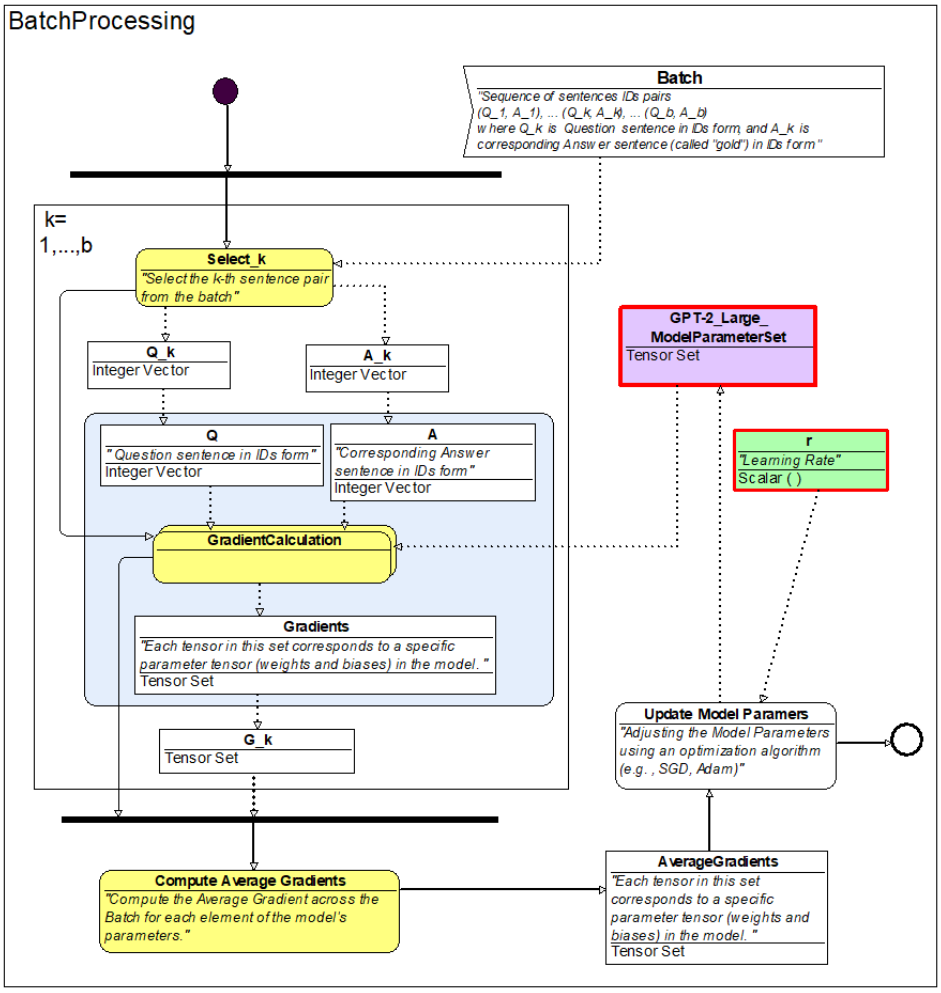


Figure FT-2

5.3. Gradient Calculation Model (Fig. FT-3)

The **Gradient Calculation Model** is responsible for calculating the gradients for the current batch. It includes the following key actions:

- **Generate Predictions**: Invokes the **Answer Generation Model** to generate predictions for the current question and calculate the loss, which is used for gradient calculation.

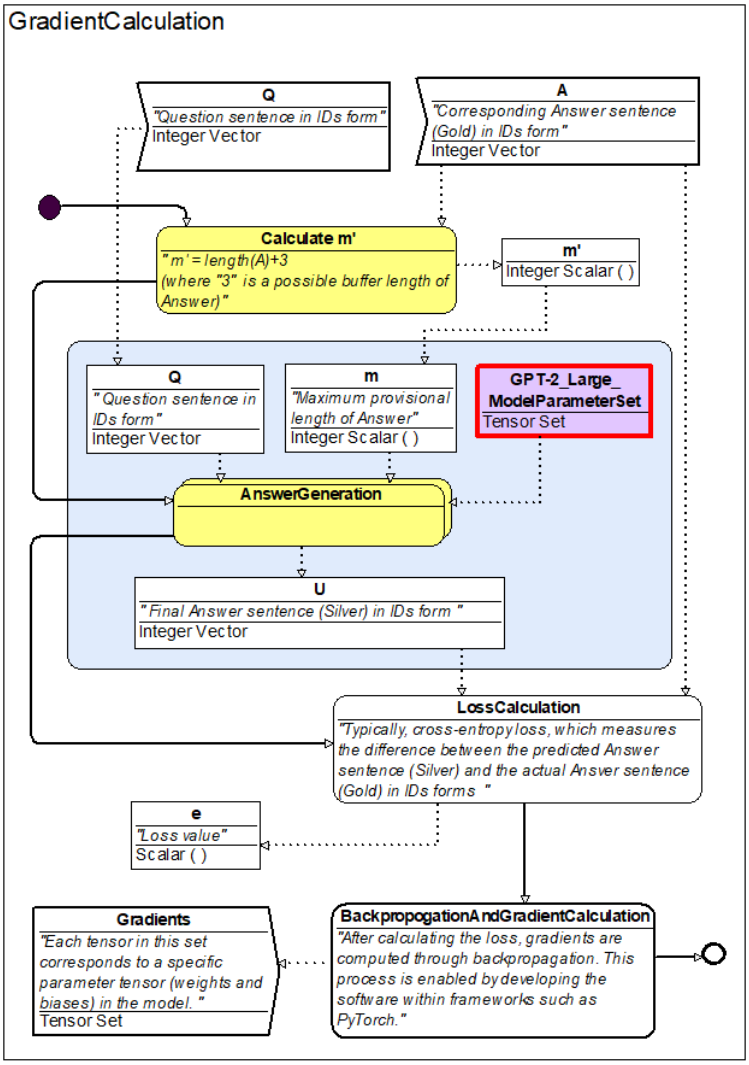


Figure FT-3

5.4. Answer Generation Model (Fig. FT-4)

The Answer **Generation Model** is responsible for processing the current query and generating a predicted answer. It includes the following key action:

- **Process Current Query**: For each question in the batch, generate the model's predicted answer.

Within the **Answer Generation Model**, during training, formatting is introduced to the sequences by adding a delimiter token, such as <|endoftext|>, to clearly mark the end of both the question and the answer. This ensures that the model knows when to stop generating responses.

- **If the model does not generate the <|endoftext|> token,** limit the generated sequence to a maximum length of n+3 tokens (where n is the number of tokens in the expected answer, with a buffer of 3).
- **Generate the answer**: The model generates an answer, stopping either when the <|endoftext|> token is produced or when the sequence reaches the n+3 limit.
- **Loss Calculation** (in Parent Gradient Calculation Model): The loss is calculated based on the tokens generated up to either the <|endoftext|> token or the maximum sequence length, whichever occurs first.

This approach ensures that the sequence generation is controlled and aligned with the expected output format.

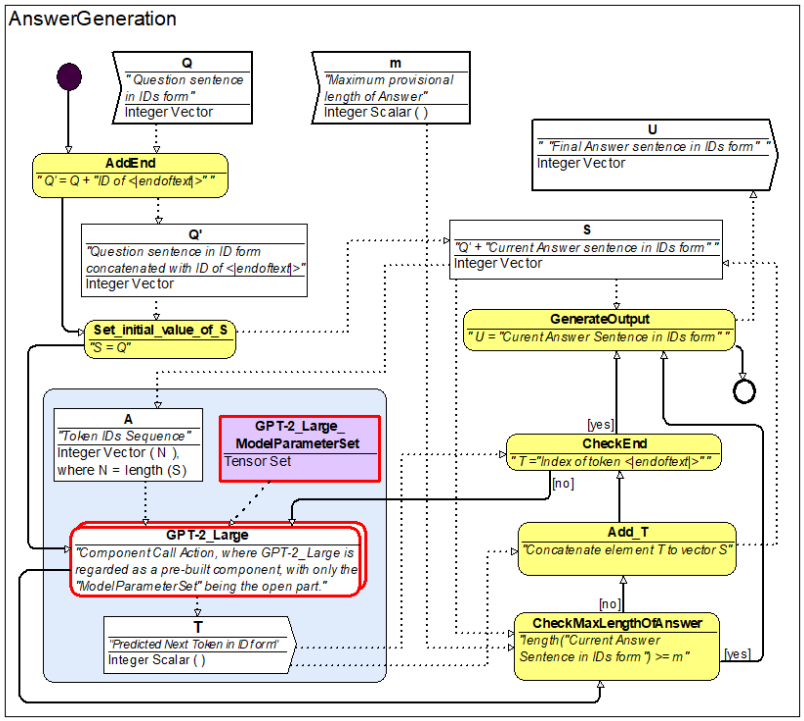


Figure FT-4

6. Conclusion

In this paper, we introduced UM1NN, a universal modeling language specifically designed for neural networks. We demonstrated its utility through two key use cases: a detailed description of the GPT-2_Large model and the fine-tuning process for question-answering tasks. These examples highlight UM1NN's capability to model complex neural network structures and operations while also showcasing its broader applicability to Machine Learning-Enabled Systems.

UM1NN goes beyond neural network modeling by providing a structured and user-friendly approach for modeling complex workflows that integrate machine learning components into larger systems. It effectively represents not only the neural networks themselves but also the surrounding systems and processes, making it a valuable tool for managing systems where machine learning plays a significant role. Its graphical representations enable clear communication among stakeholders from diverse backgrounds.

The graphical representations of UM1NN diagrams in this paper were created using the ontology graphical editor OWLGRED (Barzdins et al., 2010; WEB (g)), which supports the symbol styling mechanism needed to imitate our graphical language. Similar results can be achieved using freely available tools like Dravio (WEB (h)) and others.

Looking ahead, a potential direction for future research is to explore whether a detailed system described using UM1NN can be automatically translated into executable code with the assistance of advanced language models like ChatGPT-4 or its successors. Developing a robust serialization method for UM1NN would be crucial for this endeavor. Such a method would convert graphical representations into a format that is easily interpretable by both humans and language models, potentially enabling these models to generate implementation code based on the comprehensive system descriptions provided in UM1NN. Promising insights for this line of research are offered in recent papers (Combemale B., 2023; Petrovic N., 2023; WEB (i)), which discuss ChatGPT in software modeling.

This advancement could simplify the implementation of complex machine learning systems, making it easier for stakeholders to move from high-level design to executable code. Such automation would streamline the development process and enhance collaboration between domain experts and developers.

In conclusion, UM1NN offers a promising framework for modeling neural networks and Machine Learning-Enabled Systems. Further refinement and extension of this language could support more complex architectures and workflows, making it a practical tool for both system design and implementation in various applications.

Acknowledgments

The research was partially supported by the EU Recovery and Resilience Facility project Language technology Initiative (No 2.3.1.1.i.0/1/22/I/CFLA/002), the EU Recovery and Resilience Facility project Latvian Quantum Initiative (No 2.3.1.1.i.0/1/22/I/CFLA/001), and by research organization base financing at the IMCS UL.

References

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado GS. et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467v2, 2016

Alammar J. (2019). The Illustrated Transformer. https://jalammar.github.io/

Barzdins J., Barzdins G., Cerans K., Liepins R., Sprogis A. (2010). UML style graphical notation and editor for OWL 2. *Lecture Notes in Business Information Processing, vol. 64*, Springer, 2010, pp. 102-114

Brambilla M., Cabot J., Wimmer M. (2012). *Model-Driven_Software_Engineering_in_Practice*. Morgan & Claypool Publishers

Chollet F., Watson M. (2024). *Deep Learning with Python*, 3rd ed., Manning Publications, MEAP

Combemale B., Gray J., Rumpe B. (2023). ChatGPT in software modeling. *Software and Systems Modeling, vol.22*, 2023. https://link.springer.com/article/10.1007/s10270-023-01106-4

Farris D., Raff E., Biderman S. (2024). How GPT Works. Manning, MEAP Edition, version 7

Friedenthal S., Moore A., Steiner R. (2014). *A Practical Guide to SysML*, Third Edition: The Systems Modeling Language, Morgan Kaufmann Publishers

Friese P., Efftinge S., Köhnlein J. (2008). Build your own textual DSL with Tools from the Eclipse Modeling Project. https://www.eclipse.org/articles/Article-BuildYourOwnDSL/

Horvath A., Rath I., Varro D. (2015). Introducing EMF-IncQuery: super-fast incremental query evaluation over EMF models. https://www.eclipse.org/community/eclipse_newsletter/2015/november/article4.php

Kirchhof J.C., Moin A., Badii A., Guennemann S., Challenger M. (2022). MDE for Machine Learning-Enabled Software Systems: A Case Study and Comparison of MontiAnna & ML-Quadrat. arXiv:2209.07282v1

Kusmenko E., Nickels S., Pavlitskaya S., Rumpe B., Timmermanns T. (2019). Modeling and Training of Neural Processing Systems. In MODELS'19 (Munich). IEEE, pp.283–293

Lukyanenko R., Samuel B., Parsons J., Storey V., Pastor O., Jabbari A. (2024). Universal conceptual modeling: principles, benefits, and an agenda for conceptual modeling research. *Software and System Modeling, vol. **23***. https://link.springer.com/article/10.1007/s10270-024-01207-8

Michael J., Bork D, Wiimmer M., Mayr H. (2023). Quo Vadis Modeling? *Software and System Modeling, vol.**23***, 2023. https://link.springer.com/article/10.1007/s10270-023-01128-y

Naveed H., Arora C., Khalajzadeh H., Grundy J., Haggag O. (2024). Model driven engineering for machine learning components: A systematic literature review. *Information and Software Technology*, **169** (2024). https://arxiv.org/abs/2311.00284

OMG (2017). Unified Modeling Language. Standard, Version 2.5.1. Object Management Group (OMG). https://www.omg.org/spec/UML/2.5.1/

OMG (2024). Kernel Modeling Language (KerML), Version 1/0 Beta 2, Release 2024-02. https://www.omg.org/spec/KerML/1.0/Beta2/PDF

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703v1

Petrovic N., Al-Azzoni I. (2023). Automated approach to model-driven engineering leveraging ChatGPT and Ecore. In: 16th International Conference on Applied Electromagnetics, 2023, Serbia. https://www.researchgate.net/publication/373439135

Pires L., Guizzardi G., Wagner G., Almeida J. (2024). An Analysis of the Semantic Foundation of KerML and SysML v2. The 43rd International Conference on Conceptual Modeling (ER 2024), Carnegie Mellon University, Pittsburgh, USA. https://www.researchgate.net/publication/383738728_An_Analysis_of_the_Semantic_Foundation_of_KerML_and_SysML_v2

Radford A., Wu J., Child R., Luan D., Amodel D., Sutskever I. (2019). Language Models are Unsupervised Multitask Learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Rädler S., Berardinelli L., Winter K., Rahimi A., Rinderle-Ma S. (2014). Bridging MDE and AI: a systematic review of domain-specific languages and model-driven practices in AI software systems engineering. *Software and System Modeling.* Published online: 28 September 2024. https://arxiv.org/abs/2307.04599v2

Rumbaugh J., Jacobson, I., Booch G. (2005). *The Unified Modeling Language Reference Manua*l, Second ed., Addison-Wesley

Shapiro R., White S.A., Bock C., Muehlen M., Brambilla M., Gagne D et al. (2012). *BPMN Handbook*, Second ed., Future Strategies Inc.

Steinberg D., Budinsky F., Paternostro M., Merks E. (2009). *EMF: Eclipse Modeling Framework,* 2nd ed. Addison-Wesley Professional

Van der Aalst W., van Hee K. (2002). *Workflow Management: Models, Methods, and Systems*, MIT Press

Vaswany A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

Vaswany A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser L., Polosukhin I. (2023). Attention is all you need. arXiv:1706.03762v7

WEB (a) What is BPMN? Business Process Model and Notation. CAMUNDA. https://camunda.com/bpmn/

WEB (b). SysML Diagram Tutorial. https://sysml.org/tutorials/sysml-diagram-tutorial/

WEB (c). Keras home page. https://keras.io/

WEB (d). Netron: Visualizer for neural network, deep learning and machine learning models. https://github.com/lutzroeder/netron

WEB (e). Deep Learning Studio – ArcGIS. https://doc.arcgis.com/en/deep-learning-studio/latest/get-started/about-deep-learning-studio.htm

WEB (f). Deep Learning Studio homepage by Deep Cognition. https://deepcognition.ai/

WEB (g). OWLGrEd home page. http://owlgred.lumii.lv/

WEB (h). Draw.io tool home page. https://www.drawio.com/

WEB (i). Top 20 ChatGPT Prompts For Machine Learning. https://www.geeksforgeeks.org/top-chatgpt-prompts-for-machine-learning/

WEB (j). Loop in an activity diagram. https://arxiv.org/abs/2311.00284

Received October 27, 2024, accepted December 6, 2024

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 67-74
https://doi.org/10.22364/bjmc.2025.13.1.04

Implicit Parameter Scope Handling in Programming Languages

Mikus VANAGS

Logics Research Centre, Sterstu street 7-6, Riga, LV 1004, Latvia

mikus.vanags@logicsresearchcentre.com

ORCID 0009-0001-4542-7097

Abstract: This paper introduces a novel abstract syntax approach designed to simplify the scope and implicit parameter management in nested anonymous methods across programming languages. The proposed innovations include: 1) non-capturing function - a new method for declaring anonymous methods that does not capture implicit parameters, and 2) shorthand higher-order function call - a novel technique for invoking methods that captures implicit parameters within the scope of the function call, thereby generating a new anonymous function to be passed to the calling function. These advancements enable a more concise syntax for anonymous methods, enhancing code readability. Furthermore, the approach to implicit parameter handling in nested anonymous methods improves the conceptual understanding of boundaries and interactions between complex nested anonymous functions. Collectively, these innovations pave the way for more intuitive, maintainable, and expressive anonymous methods in programming languages.

Keywords: programming languages, implicit parameter, anonymous methods, parameter scope.

1. Implicit parameters

Scala language has feature named ‘contextual parameters’ aka ‘implicit parameters’ (WEB, a) which is something between global variables and default arguments rather than feature completely enclosed inside the method body declaration as are method parameters. Kotlin® supports keyword ‘it’ (WEB, b) – it can be used in lambda expressions as single parameter with constant name which might affect code readability. Q language supports up to 3 implicit parameters with special names x, y and z (WEB, c) which also is a limitation of the expressiveness and might affect code readability. Swift® has shorthand argument names (WEB, d) which allows to refer to lambda parameter using index of the parameter - that is similar idea to Clojure shorthand lambda syntax (WEB, e) - to use indexes instead of names which also might affect code readability. Accessing parameters by indexes or using constant names to access the parameters is ambiguous in nested lambda expressions, therefore is needed better, more abstract and more expressive model of implicit parameters (Vanags and Cevere, 2018).

The idea of implicit parameters redefine how parameters are declared in programming, moving the parameter declaration from the method's signature to the body of the method. In this approach, all unknown identifiers within the method body are treated as parameters that have been implicitly declared. These implicit parameters are handled as expressions,

and the first occurrence of an implicit parameter expression leads to its addition to the list of method parameters (Vanags et al., 2016). In the following pseudocode example parameter 'x' is implicitly defined:

```
function {return x+6;}
```

Following pseudocode demonstrates equivalent example declaring parameter explicitly (the usual way how it is done in programming languages):

```
function(x) {return x+6;}
```

Implicit parameters make method declarations more concise, thereby improving code readability, particularly for lambda expressions (relatively small and simple code expressions), making this approach potentially applicable across various programming languages. Comparison of possible anonymous method syntax improvements related to implicit parameters are shown in Figure 1 demonstrating how implicit parameters facilitate removing unnecessary keywords and symbols from lambda syntax making the lambda syntax very concise.

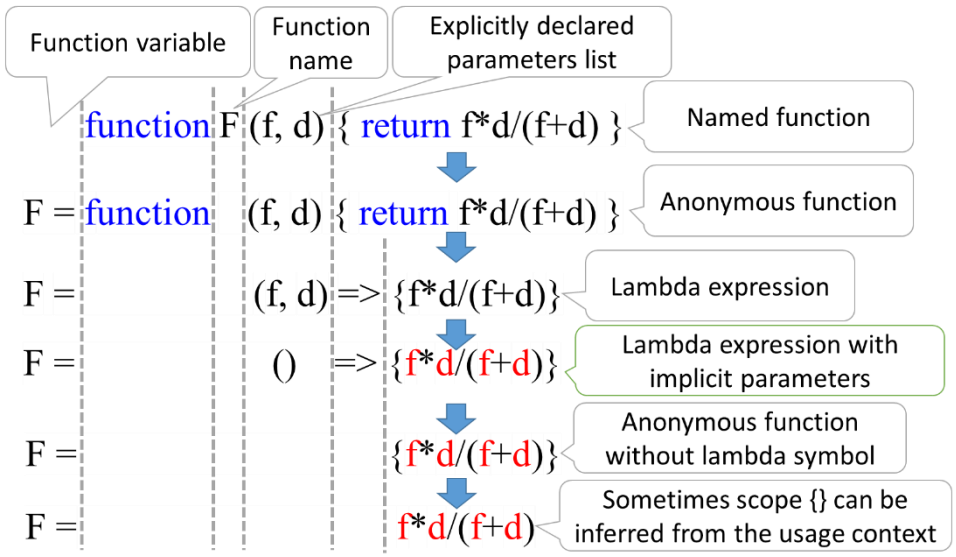


Figure 1. Anonymous method syntax improvements using implicit parameters.

Implicit parameters are placed in the method's parameter list in the same order as they are identified within the method body. The parameter list is part of the method's signature, and there are occasions when altering the order of these parameters is desirable. Implicit parameter order can be changed using Grace~ operator (Vanags, 2016). The prefix form of the Grace~ operator shifts a parameter one position closer to the beginning of the method's parameter list:

```
function {return y - ~x;}
```

Conversely, the postfix form of the Grace~ operator shifts a parameter by one position towards the end of the method's parameter list:

```
function {return y~ - x;}
```

Both Grace~ operator usage examples are functionally equivalent to the following example without using implicit parameters:

```
function(x, y) {return y - x;}
```

Currently only KatLang programming language (WEB, f) implements implicit parameter feature.

2. Implicit Parameters in Nested Functions

In the following ECMAScript aka JavaScript (WEB, g) example (using explicitly declared parameters) parameters a and b belong to the function assigned to the variable f. In the case of the anonymous function assigned to the function variable g, the parameter c is associated with the nested anonymous function.

```
var f = function(a, b) { return a(b) }
var g = function() {
    return f(function(c) {return c+1}, 2)
}
g()
```

Functionally equivalent example using implicitly declared parameters is as follows:

```
f = function { return a(b) }
g = function {
    return f(function {return c+1}, 2)
}
g()
```

More concise syntax can be achieved by removing all the unnecessary keywords:

```
f = { a(b) }
g = {
    f({ c+1 }, 2)
}
g
```

In a nested functions environment, each implicit parameter is captured by the most nested function within which the implicit parameter is encountered for the first time. The outer anonymous function does not contain any parameters because implicit parameter c is first encountered and thus captured within the nested anonymous function.

It is important to note that an implicit parameter will invariably be captured by some function. If not captured by any inner function, the last opportunity for capture is by the most outer function. Consequently, the most outer function is always defined with {} brackets, but since these brackets are a constant feature of the outermost function, they can be omitted and inferred from the context of the code's use.

Previous example can be improved by removing unnecessary brackets as follows:

```
f = a(b)
g = f({ c+1 }, 2)
g
```

When all the unnecessary symbols and keywords are removed, it is a little easier to notice which anonymous function captures which implicit parameters. The result is KatLang example. KatLang allows for the omission of the outermost function's brackets {} because it interprets line endings as possible ending of an expression. While such a practice may not align with the syntax of all programming languages, it illustrates the potential for making code more concise by eliminating unnecessary constants, symbols, or keywords.

3. Grace~ operator role in changing the scope of implicit parameters

Implicit parameters are limited to the scope in which they are defined, but Grace~ operator can be used to change the scope of an implicit parameter. It means that Grace~ operator prefix form can move implicit parameters to outer scope as demonstrated in following example:

```
f = function{
    return function {
        return ~a+1
    }
}
```

The same example without unnecessary keywords:

```
f = {
    {~a + 1}
}
```

Parameter a is tried to move one position before the beginning of the parameters list and it is interpreted as moving the parameter to the end of the outer function parameters list.

Functionally equivalent JavaScript example using explicitly declared parameters are as follows:

```
var f = function(a) {
    return function() {
        return a + 1
    }
}
```

The postfix form of Grace~ operator does not have capability to change the scope of implicit parameter, therefore prefix and postfix forms of the Grace~ operator is asymmetric.

Symmetry, simplicity and code readability are the reasons why Grace~ operator is better to be limited to work only in the scope of single function and do not allow to move

the parameter outside the visibility scope of the function. A better solution is needed to control the scope of implicit parameters.

4. Non-capturing function – parameter less function with implicit parameters in the function body

For the sake of simplicity, the function which can contain parameters is called a 'parametrized' function. In the world of implicit parameters, the only way for the function to be parameter less function is not to have any implicit parameter in the body of the function. But that is a serious limitation for the language expressiveness. The limitation can be overcome by creating non-capturing function - a new kind of function which is defined between brackets '(' and ')' and such function does not own or capture any implicit parameter. The implicit parameters are owned by the closest parametrized function and the parametrized function is defined between brackets '{' and '}'. The most outer function is parametrized by default, because some function needs to capture implicit parameters if they are not captured in the inner functions. Therefore, in the definition of the most outer function bracket { } usage is optional.

Figure 2 explains how JavaScript syntax can be improved demonstrating the usage of parameter less function defined withing brackets ( ).

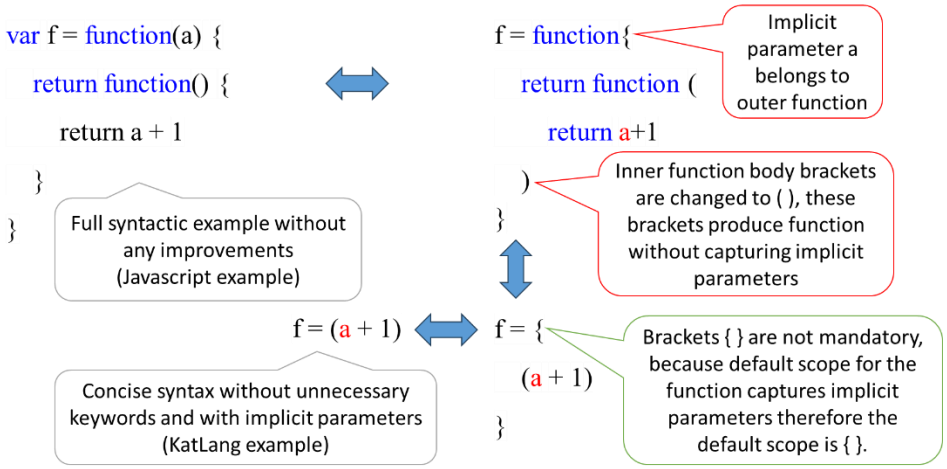


Figure 2. Deduction of parameter less function syntax with explanations.

Function f=(a+1) contains outer function (which is parametrized by default) and inner function defined with brackets ( ). Usually in programming languages it means that the function can be executed and then the resulting function (returned from the initial call) can be executed. The execution syntax is following: f(1)(). To simplify syntax for such cases when the function is called, the execution can be performed to all the returned (inner) parameter less functions until no more executions are possible. It means unwrapping the nested (inner) parameter less function which results in getting rid of the unnecessary

brackets (scopes). Such behavior might not be the best in all possible situations, but for processing math expressions, it makes sense and works well in programming language KatLang.

5. Shorthand higher order function call

If the function body can be declared in two different ways – using brackets () or {} as shown in the previous chapter, then the same principle can be applied to method calls which means implementing the method call using brackets {} instead of (). It means the method call will be specialized to pass lambda expression as parameter to the calling method. Figure 3 shows how to convert explicit parameters example to implicit parameters example, then how to make syntax more concise by removing unnecessary keywords and brackets, and then how to improve syntax by using brackets {} in the method call. This approach simplifies the syntax for shorthand higher-order function calls by allowing a lambda expression to be passed to a function that expects another function as a parameter.

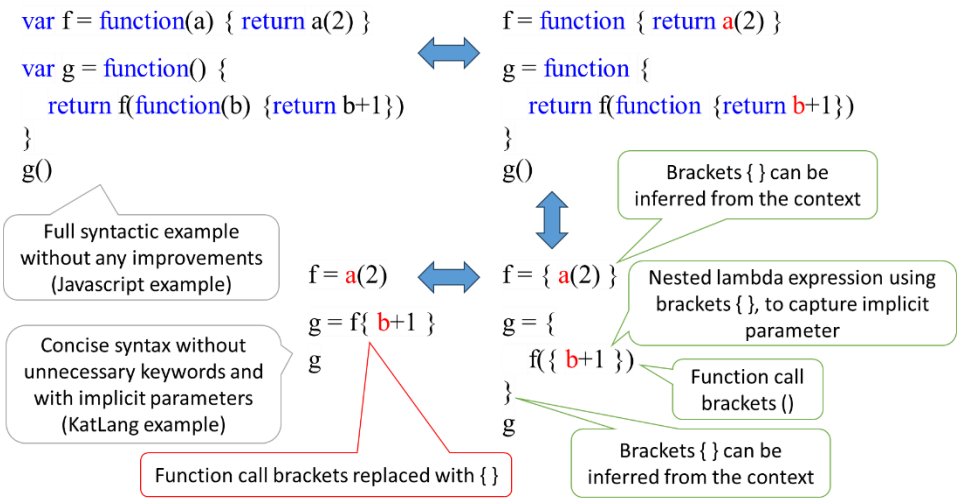


Figure 3. Deduction of shorthand higher order function call syntax with explanations.

Bracket {} usage in method call expressions is quite simple – the argument of the method call is a function which body is defined between brackets {}. More interesting now seems bracket () usage in method calls. When brackets () are used for the method call, it can be interpreted as passing parameter less function to the method call. Any parameter less function, including nested parameter less functions, can be unwrapped. Usually programming languages allow passing the final unwrapped expressions of the parameter less function to the method call – it is because in other programming languages brackets () are used to manage the operation priorities without creating a new lambda expression. Parameter less function is like a wrapping layer to the unwrapped expressions

and structurally it is similar to the scenario when the parametrized function is passed to the method call.

Parameter less function unwrapping takes processor time, and it might not be the best option for all the programming languages, but nonetheless it is an interesting concept worth considering when designing future programming languages.

6. Summary

This paper introduces two key innovations to improve the handling of implicit parameters in programming languages: the non-capturing function and the shorthand higher-order function call.

The non-capturing function - defined with parentheses () - allows for the inclusion of implicit parameters without capturing them within the function. Instead, implicit parameters are bound to the nearest outer parametrized function, defined with curly brackets {}. This distinction reshapes traditional approaches to function scopes and parameter management, shifting the way developers can think about function definition, parameter binding, and scope visibility.

The Grace~ operator provides a method for reordering implicit parameters within the method's parameter list, but its prefix and postfix forms are asymmetric in controlling parameter scope. While the operator is effective in reordering parameters, it is less ideal for managing the parameter scope in nested functions. The non-capturing function offers a more robust solution to this challenge by separating the logic of the function from parameter ownership.

Additionally, the shorthand higher-order function call simplifies function invocation by allowing the omission of traditional function call brackets () when passing a lambda expression as an argument. This concise syntax enhances readability and reduces unnecessary symbols, particularly in functional programming contexts.

Both innovations have been implemented in the abstract programming language for math calculations - KatLang, which supports implicit parameters and the Grace~ operator. KatLang challenges traditional paradigms of function declaration and usage, encouraging developers to rethink the relationship between a function’s logic and its parameter interface. These contributions represent a conceptual shift that enhances readability, clarity, and expressiveness in programming language design.

References

Vanags, M., Justs, J., Romanovskis, D. (2016). Implicit parameters and implicit arguments in programming languages. US Patent 9361071.

Vanags M. (2016). Grace~ operator for changing order and scope of implicit parameters. US Patent 9417850.

Vanags M., Cevere R. (2018). The Perfect Lambda Syntax. Baltic J. Modern Computing, Vol. 6 (2018), No. 1, 13-30 Retrieved from https://doi.org/10.22364/bjmc.2018.6.1.02

WEB (a). Tour of Scala, Contextual parameters, aka implicit parameters. Retrieved May 5, 2024, from https://docs.scala-lang.org/tour/implicit-parameters.html

WEB (b). Kotlin, Higher-order functions and lambdas (2023). Retrieved May 5, 2024, from: https://kotlinlang.org/docs/lambdas.html#it-implicit-name-of-a-single-parameter

WEB (c). Kdb+ and q documentation, Function notation. Kx Systems, Inc. (2023). Retrieved May 5, 2024, from https://code.kx.com/q/basics/function-notation/

WEB (d). The Swift Programming Language (5.10), Shorthand Argument Names (2023). Apple Inc. Retrieved May 5, 2024, from https://docs.swift.org/swift-book/documentation/the-swift-programming-language/closures/#Shorthand-Argument-Names
WEB (e). Clojure, Reader, Hickey R. (2022). Retrieved May 5, 2024, from https://clojure.org/reference/reader
WEB (f) KatLang. Logics Research Centre (2022). Retrieved May 5, 2024, from http://katlang.org/
WEB (g) ECMAScript® 2023 language specification (2023). Retrieved May 5, 2024, from https://ecma-international.org/publications-and-standards/standards/ecma-262/

Received May 11, 2024, revised September 22, 2024, accepted January 13, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 75-95
https://doi.org/10.22364/bjmc.2025.13.1.05

Mobile Device-Based Ants Recognition and Tracking System: Methodology and Frameworks

Dmytro KUSHNIR

Department of Computer Engineering, Lviv Polytechnic National University,
Stepana Bandery 12, Lviv, 79013, Ukraine

dmytro.o.kushnir@lpnu.ua

ORCID 0000-0001-6623-3382

Abstract. This research introduces a practical framework for ex situ ants recognition and tracking, enabling the continuous monitoring of their movements. This framework includes an iOS-based client application with an embedded Visual Intersection Over a Union (V-IOU) tracking module and a You Only Look Once (YOLO) recognition model. Another part of a framework is a scalable system for autonomous annotating, training, and converting the model to mobile format. The tracking algorithm is integrated into a Swift application using the JavaScriptCore library, while the Yolo model is integrated using the CoreML framework. To ensure system accuracy and stability, the method of improved clasterization K-means++ is employed. Simultaneously, the Affine Quantization method is used keep the model size as small as possible. An experimental benchmark on an indoor ant colony was conducted, using various recognition models to assess the accuracy and productivity of the system. The results confirm the practicality of our methods and frameworks for real-time small object detection, demonstrating their applicability in real-world scenarios.

Keywords: Affine Quantization, Ants Tracking, K-means++ Clustering, CoreML iOS Framework, JavaScriptCore Library, Object Detection, Real-time Tracking, Scalable System, V-IOU Tracking, YOLO Model

1. Introduction

The study of insect behavior, particularly ants, has long fascinated researchers due to these tiny creatures' complex social structures and behaviors. Accurate tracking and recognition of ants can provide invaluable insights into their collective behavior, foraging patterns, and colony dynamics (Popp et al., 2024). Beyond ecological curiosity, understanding ant movements and interactions can show how resources are distributed within colonies and help identify patterns in their decision-making and organization.

While studying ants in their natural habitat is important, this research focuses on ex-situ monitoring in controlled settings. By observing ants in a controlled environment, the study aims to systematically track their movements, identify areas of high activity, and analyze behavioral trends such as trail formation and resource

allocation. These insights are difficult to obtain through traditional methods, often relying on labor-intensive manual observations or non-scalable technologies.

To address these challenges, recent advancements in object detection and tracking technologies provide promising solutions. The You Only Look Once (YOLO) model, known for its high-speed and accurate object detection capabilities, has been successfully applied in various fields, including wildlife monitoring and urban surveillance. Similarly, tracking algorithms such as the Visual Intersection over Union (V-IOU) (Bochinski et al., 2018) tracker have shown promise in maintaining the identity of moving objects over time. Building on these technologies, this research aims to develop a practical, scalable solution for continuous ant monitoring by integrating these state-of-the-art methods.

1.1. Purpose of the study

The primary purpose of this study is to develop a practical and scalable framework for continuously monitoring ants using a combination of the V-IOU tracking module and the You Only Look Once (YOLO) recognition model. This framework is implemented on an iOS platform, making it accessible and user-friendly for field researchers and enthusiasts. By integrating these advanced technologies, the study aims to provide an efficient and accurate solution for real-time ant tracking, significantly reducing the need for manual observations.

1.2. Contributions of the study

This study makes several key contributions to the field of entomology and computer vision:

Innovative Framework: Introduction of a comprehensive iOS-based framework that integrates V-IOU tracking and YOLO recognition for real-time monitoring of ants.

Advanced Integration: Implement the tracking algorithm within a Swift application using the JavaScriptCore library and the embedding of the YOLO model using the CoreML framework.

Optimization Techniques: Improved clustering methods (K-means++) and Affine Quantization enhance system accuracy and stability while keeping the model lightweight and responsive.

Scalability: Development of an autonomous system capable of annotating, training, and converting models for mobile deployment, ensuring adaptability to various environments.

Experimental Validation: Experimental benchmarks were conducted on an indoor ant colony to validate the framework's accuracy and productivity and demonstrate its practicality in real-world scenarios.

1.3. Organization of the study

The remainder of this paper is structured as follows: Section 2 presents the literature review, offering an overview of insect tracking and recognition technologies research. Section 3 details the methodology, including the design and implementation of the

recognition model, its training, clusterization, and the integration of the tracking method. Section 4 discusses the frameworks used, focusing on the scalable system for autonomous annotating, training, and converting the recognition model to a mobile format. Section 5 describes the experimental setup and results, including the benchmark conducted on an indoor ant colony and the results obtained. In Section 6, the discussion analyzes the findings, compares them with existing solutions, and explores the study's implications. Finally, Section 7 provides the conclusions, summarizing the study's contributions, suggesting areas for further research, and recommending improvements for future implementations.

2. Related Work

Recent research on car traffic monitoring using UAVs conducted by Gudauskas et al. (2024) demonstrated the importance of quick custom object recognition and tracking in real-time environments. Similarly, advancements in insect monitoring technologies have enabled real-time tracking and behavioral analysis of various species. For instance, the IntelliBeeHive system integrates machine learning and computer vision to monitor honeybee activity, detect pests, and provide insights to beekeepers (Smith et al., 2023). The AROBA system further highlights the use of autonomous observation technologies for honeybee colonies, emphasizing continuous monitoring without human intervention (Ulrich et al., 2024). These innovations demonstrate the potential of leveraging advanced tools for studying insect behavior and health, paving the way for real-time, scalable monitoring solutions.

Building on these foundations, this study focuses on recognizing and tracking ants using mobile device-based systems. While honeybee monitoring often relies on fixed systems or specialized setups, my approach targets a more flexible and portable framework that can be adapted for recognizing and tracking various objects, including but not limited to ants. By implementing the YOLO model (Bochkovskiy et al., 2021) for efficient real-time object detection and the V-IOU algorithm (Bochinski et al., 2018) for robust tracking, the framework offers a novel solution tailored to ants' unique behaviors. This aligns with previous research on small dynamic object recognition, such as that conducted in my Ph.D. thesis (Kushnir, 2023), which emphasized the challenges of adapting such systems for mobile platforms.

On the other hand, integrating such recognition and tracking systems on embedded devices, as I did in the previous research (Kushnir, 2022), creates a high load on the Graphical Processing Unit (GPU), which, in theory, can be improved by using a Neural Processing Unit (NPU) from a mobile device. Also, it is worth noting that previously implemented recognition and tracking algorithms executed on separate Docker environments should be migrated to mobile devices like iOS without losing efficiency.

To resolve such issues, a practical approach involves utilizing the k-means++ clustering algorithm proposed by Arthur et al. (2007) on the YOLO recognition model to divide recognized clusters of objects into corresponding classes correctly. This is vital in the scope of multiple small object recognition, like ants, which can move fast and fit on each other. Research conducted by Wang et al. (2023) with an improved VV-YOLO model confirms such assumptions, showing an improved real-time vehicle

recognition process.

Additionally, it is crucial to minimize recognition model weights on mobile devices without sacrificing efficiency. As Li et al. (2023) demonstrated, model quantization methods can achieve this, where a fully quantized network with 4-bit quantization showed an acceptable accuracy loss.

CoreML tools, as presented by Marques (2020), were utilized to integrate the model into an iOS mobile device. These tools facilitate the seamless conversion and deployment of machine learning models onto iOS platforms, ensuring efficient performance and leveraging the advanced hardware capabilities of Apple's devices. By using CoreML, the model can take advantage of on-device processing, which enhances speed and privacy by minimizing the need for data to be sent to external servers. This integration is particularly beneficial for applications requiring real-time processing so that it can be used for ant processing.

For tracking module injection, I propose using the JavaScriptCore Swift framework analyzed by Novák (2020). This framework allows for the seamless integration of JavaScript code within Swift applications, enabling efficient execution and manipulation of JavaScript within the iOS environment in real-time. That can be achieved by creating JSContext for each recognition thread in the Swift application, efficiently increasing real-time tracking.

The decision to integrate the tracking algorithm using JavaScriptCore was motivated by its capacity to support a cross-platform implementation strategy. JavaScriptCore allows the tracking logic, written in JavaScript, to be readily adapted for deployment in various environments, including web servers and embedded systems, thereby enhancing the scalability and versatility of the proposed framework. Additionally, this choice facilitates modularity by decoupling the algorithm's implementation from the iOS application, ensuring its reuse across multiple platforms without significant modifications. In the context of this research, JavaScriptCore enabled precise and reproducible experimentation in a controlled indoor setting, while its dynamic runtime capabilities allowed iterative fine-tuning of the tracking logic.

3. Methodology

3.1. General workflow

To fulfill the research goals, a system for autonomous annotating, training, and converting the Recognition model to CoreML Mobile format is proposed (Figure 1).

This system receives datasets of images annotated either automatically or manually for specific classes of objects, trains the model, and converts it to the CoreML format. On the client side, an iOS Swift application retrieves the converted model and the necessary metadata. The tracking module is also integrated using the JavaScriptCore framework to facilitate real-time tracking of the required object classes. Each system module will be discussed in detail in the “Frameworks” section of the article.

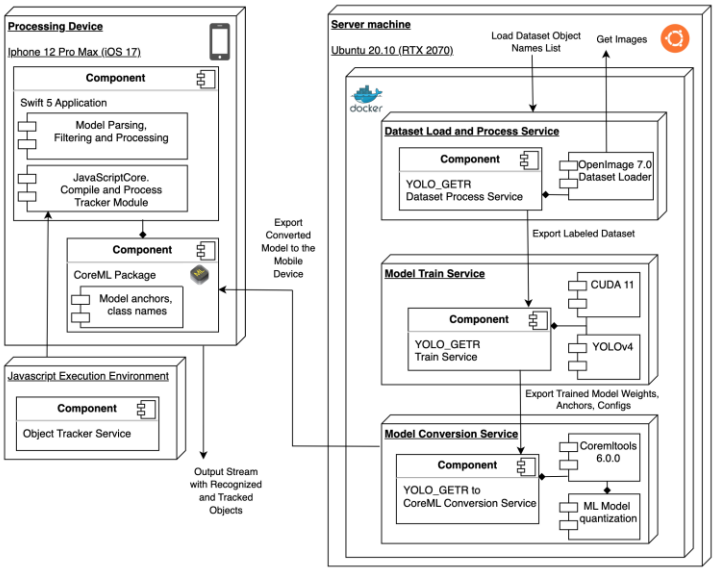


Figure 1. General workflow of the system

This system receives datasets of images annotated either automatically or manually for specific classes of objects, trains the model, and converts it to the CoreML format. On the client side, an iOS Swift application retrieves the converted model and the necessary metadata. The tracking module is also integrated using the JavaScriptCore framework to facilitate real-time tracking of the required object classes. Each system module will be discussed in detail in the "Frameworks" section of the article.

3.2. Dataset

The dataset used in this research is formed from two sources: a manually labeled indoor ants dataset and an automatically generated dataset from Open Images for a specified object class. These two datasets train the YOLOv4 model (Bochkovskiy et al., 2021).

The manually labeled dataset was created using the open-source tool LabelStudio (Tkachenko et al., 2024), which allows quickly annotating multiple small objects on the image frame, which is vital for ants labeling. This resulted in a dataset comprising 500 images (Kushnir, 2022).

Conversely, the autonomous dataset was created for a specific class of ants using OpenImage 7.0. This process involved forming a list of input classes and downloading the required number of annotated images and the necessary metadata for the recognition model. Com bining the automatically downloaded dataset with the manually labeled one enhances the diversity of the annotated images, which is expected to positively impact the model weight creation during the training process.

Ultimately, this approach increased the total number of annotated images to 1500, significantly enhancing the dataset for more robust model training.

3.3. Recognition model and evaluation metrics

Among the Convolutional Neural Networks (CNN) researched in my Ph.D. thesis (Kushnir, 2023), YOLOv4 was proposed as the recognition model due to its efficiency and accuracy in real-time object detection. The model's architecture, including the backbone and neck, remained unchanged. Two configurations were tested: a "tiny" model with two output layers and a larger model with three output layers. The RETR model represents a custom-created model designed for "Recognition and Tracking". The hyperparameters for these models are detailed in Table 1.

Table 1. Parameters used in model training

CNN Model	Batch	Subdivisions	height /width	Learning Rate (LR)	Decay	Momentum
Yolov4_retr	64	16	416/416	0.002	0.0005	0.95
Yolov4_retr_tiny	64	8	416/416	0.001	0.0005	0.9

The gradient descent algorithm used for optimizing hyperparameters was Nesterov Accelerated Gradient (NAG), utilizing specific values for Decay and Momentum. The loss function employed, particularly for the localization component, was the Complete Intersection Over the Union (CIoU) method. This approach minimizes the normalized distance between two analyzed objects, enhancing detection accuracy.

Several metrics were proposed to benchmark recognition results during the research to evaluate the model's performance. Among these, confusion matrix metrics are particularly important for recognition tasks using supervised learning and imbalanced classes. This method categorizes recognition objects into four categories based on the combination of positive response and algorithm: true positive (**TP**), true negative (**TN**), false positive (**FP**), and false negative (**FN**).

Let's introduce the intersection over union (IoU) minimization filter to better understand how these values are calculated.

$$\boldsymbol{IOU}(\boldsymbol{A},\boldsymbol{B}) = \frac{A \cap B}{A \cup B}$$

This metric determines how much the recognition region and the reference object overlap in internal volume. A recognition result is considered a true positive (TP) detection if the IoU equals or exceeds 0.5. An FP state occurs when the IoU values are below 0.5. FN is a state where true positive objects were not detected or were below the set threshold. FP defines a state where false negative results are interpreted as positive.

Based on that, the following metrics were applied:

$$\boldsymbol{R}\ (recall) = \left(\frac{TP}{TP + FN}\right)$$

$$\boldsymbol{FNR}\ (false\ negative\ rate) = 1.0 - R$$

$$\boldsymbol{P}\ (precision) = \left(\frac{TP}{TP + FP}\right)$$

$$\boldsymbol{FPR}\ (false\ positive\ \ rate) = 1.0 - P$$

Recall indicates the proportion of TP objects identified by the classifier. Precision indicates the proportion of objects identified as TP that are truly positive. FPR indicates the expected duration of FP states. FNR represents the fraction of all FNs that still yield positive results. The lower the values of FNR and FPR, the higher the model's performance.

It is important to note that recall and precision are independent of the input class size ratios. If the proportion of TP objects is significantly smaller than the number of TN class objects, these metric indicators will show the correct functioning of the tested algorithms.

There are two main ways to obtain a single quality criterion using recall and precision: the F-measure and the average precision score.

$$\boldsymbol{F1} = 2\ \times \left(\frac{P\ \times\ R}{P\ +\ R}\right)$$

The feature of obtaining the harmonic mean for the F-measure is that such a measure is close to zero. Thus, higher metric accuracy is achieved with incorrect sample distribution.

However, obtaining the harmonic mean for the F-measure often results in a measure close to zero, which achieves higher metric accuracy when the sample distribution is incorrect. When it is necessary not only to predict an object's class but also to perform ranking–solving object recognition tasks of search and localization, the mean Average Precision (mAP) metric is used; this metric is vital for calculating the average classification indicators across all categories (Kushnir, 2023).

$$\boldsymbol{mAP}\ = \frac{\sum_{i=1}^{n} AP_i}{n}$$

The mAP metric determines the level of confidence for recognition objects. Therefore, this method is appropriate for assessing the effectiveness of a CNN model during training. Additionally, it can be used to compare the implemented YOLO models at the inference stage.

It is important to note that real-time GPU, CPU, and NPU load performance metrics must be added for recognition tasks on mobile devices with limited hardware capabilities.

Let us introduce concepts that define real-time performance: **FLOPs** (number of floating-point operations per second), **FPS** (frames per second), the display time of results on the screen after the start of processing – $\boldsymbol{T_{frame}}$, the median time for performing an object prediction operation – $\boldsymbol{T_{predict}}$, and the median time for loading the model onto the tested device – $\boldsymbol{T_{load}}$.

Thus, to achieve the research goals, it is necessary to determine the inference speed of the developed recognition model on mobile devices, ensuring it is close to real-time at approximately 24 FPS. Additionally, it is essential to assess the usage of CPU, GPU, and other acceleration devices using metrics such as FLOPs, $\boldsymbol{T_{predict}}$, $\boldsymbol{T_{frame}}$, and $\boldsymbol{T_{load}}$.

3.4. K-means++ clustering during model training

Training the developed CNN model can be effectively divided into two stages. The first stage uses the available hardware to generate the model's primary weight coefficients. During this stage, particular attention is given to applying the clustering method to calculate recognition anchors and set input hyperparameters. These include the number of training iterations, the determination of block and sub-block sizes of the CNN, and additional parameters based on the selected optimization algorithm. A crucial part of this stage is determining the LOSS value, which significantly impacts the model's performance.

In the second stage, the trained CNN model is fine-tuned by verifying whether the current weight coefficient with the best mAP value has reached its highest value.

The original YOLO uses the k-means unsupervised clustering method to form recognition anchors (anchor boxes), a technique for dividing the input image into a grid of cells to which the object recognition region is attached. This method determines the position, width, and height of the object relative to the center of the grid cell, using k-means to identify the most optimal anchor sizes.

The principle of k-means involves iteratively using the Euclidean distance to calculate the distance between a manually set number of clusters. During the expectation phase, the distance is calculated from the initial cluster center (centroid) to the center of each object.

A disadvantage of this algorithm is the need to know the number of clusters in advance; the result of clustering and execution time – O(n) depends on the choice of initial centroids. If the initial centroids are chosen randomly, it may lead to convergence errors.

Therefore, the modified k-means++ algorithm is proposed for generating recognition anchors. This algorithm addresses the problem of random centroid placement. The algorithm prioritizes points at the maximum distance from the centroid to avoid overlapping two points.

A test with 2000 random data points was created on four separate clusters to compare the evaluation of the two clusters (Figure 2 and Figure 3) (Kushnir, 2023).

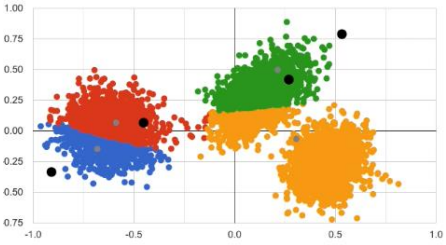


Figure 2. K-means clustering

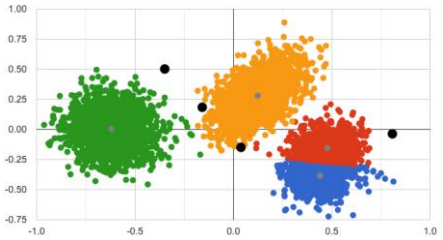


Figure 3. K-means++ clustering

Black dots on the images represent the starting coordinates of centroids, while grey dots represent the central coordinates of the clusters. As seen from the analysis results, the current organization of initial centroids in K-means clustering could have been

more successful, as data from one cluster intersected with data from another. However, with the k-means++ algorithm, points in the clusters are distributed correctly. It is worth noting that although convergence errors are significantly minimized, this comes at higher computational costs than the k-means algorithm.

Let us define the steps to apply to our YOLO model with k-means++ clustering integration, accommodating any output layer number (Algorithm 1).

Algorithm 1. Clustering using K-means++ for forming Anchors in YOLOv4

1. Determine the height and width of recognition rectangles from all recognition regions. Set the initial iteration i = 0;
2. **Repeat** iteration i;
3. Select a recognition anchor as the initial point (centroid) of the input cluster from all recognition regions;
4. Calculate the distance $D(x_i)$ between the centroid of all recognition regions and the centroid of existing recognition anchors. Calculate the probability $P(x_i)$ for each recognition region selected as the next centroid:

$$x \in X, \qquad P_i = \frac{D(x_i)^2}{\sum_{i=1}^{n}(x_i)^2}$$

The further the recognition region is from the initial centroid, the higher the probability of its selection;
5. Use the IoU minimization filter for each region and recognition anchor to select the most likely recognition anchors for the current recognition region. The higher this value, the more likely the object belongs to the desired class;
6. **Repeat** until the recognition regions do not change;
7. Obtain the final recognition anchors.

To summarize, for recognition tasks during model training, the k-means++ algorithm increases the accuracy of anchor selection, which can significantly improve error determination in image classification.

3.5. Tracking method and module injection

This study has selected the V-IOU tracking method, which I thoroughly tested in my previous research (Kushnir, 2022). On the other hand, in the current study, the JavaScriptCore framework, operating within an isolated virtual JavaScript environment, was employed to integrate this tracking method into the mobile Swift application. However, since JavaScript runs in a single thread (event loop), it imposes limitations on hardware performance. While this may be sufficient in a web browser, achieving high performance under heavy CPU and GPU loads is critical for iOS systems.

To address this issue, a multithreaded approach in Swift 5 was proposed. Each thread was given shared access to the JavaScriptCore instance (JsContext) and a separate asynchronous message queue, allowing tracking tasks to be distributed across multiple threads. Additionally, caching identical objects helps reduce the impact of the tracking process on performance.

The proposed algorithm outlines the process of initializing a JsContext instance and using it as a separate module for object tracking. Key steps include setting a batch limit for processing requests, initializing the JsRunner class, creating a shared global JsCore context, and asynchronously processing data through the tracking module.

To optimize the integration of modules on resource-constrained mobile platforms, the module size was minimized using tools like Rollup. JavaScript supports several module design patterns, and the UMD (Universal Module Definition) pattern was selected for this study. This pattern operates across various platforms, ensuring that each generated UMD module functions in an isolated environment and can be successfully integrated into the iOS mobile platform.

4. Frameworks

4.1. Scalable system for model annotation, training, and converting to mobile format

Achieving scalability and efficiency is crucial in developing machine learning models, especially for mobile platforms. Containerization presents a robust solution to meet these needs by isolating various system components through virtualization tools like Docker. This method enables modularization, allowing each container to function in its independent environment, ensuring flexibility, ease of deployment, and separation from other containers. As a result, the object recognition system crafted for mobile platforms is divided into distinct services within Docker containers, each responsible for a specific stage in the neural network model lifecycle: data annotation, model training, and model conversion to a mobile-friendly format. The proposed structural diagram of this system is illustrated in Figure 4.

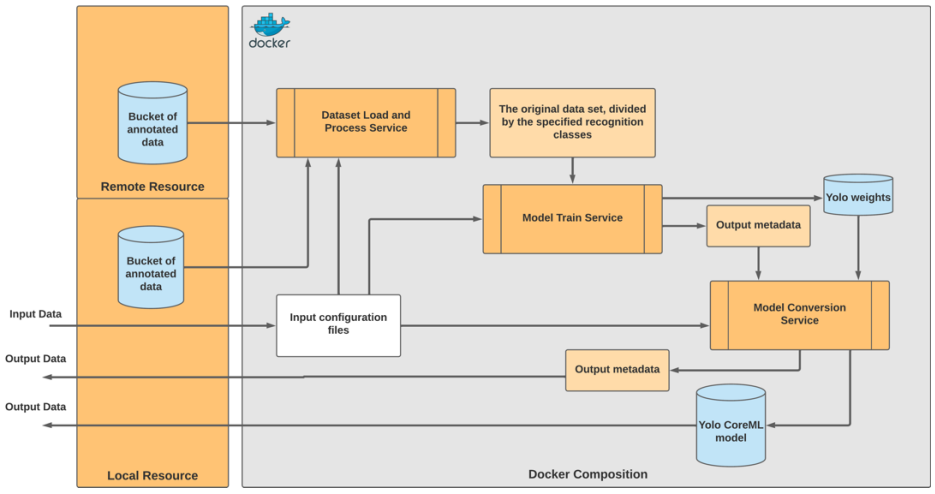


Figure 4. Structural scheme for autonomous data annotating, model training, and converting to mobile format system.

To ensure scalability, the framework supports simultaneous processing of multiple datasets by running several instances of the relevant containers. For instance, datasets can be annotated, trained, and converted concurrently, enabling efficient use of computational resources and faster model preparation. As a result, the system allows users to generate the required model weights simply by specifying the names of the classes to be trained. For example, classes such as '*ant-messor-structor'*, '*ant-camponotus-fellah'*, and others can be easily defined and processed, ensuring flexibility in handling diverse datasets.

Creating a resulting neural network for a mobile platform can be broken down into three core stages.

4.1.1. Annotation service

This service identifies and processes input datasets according to the specific recognition classes the neural network requires. It automates the loading and annotation of training and testing datasets from public databases such as OpenImage 7.0 (Figure 5). The service outputs annotated classes for each image in a format suitable for YOLO model training.

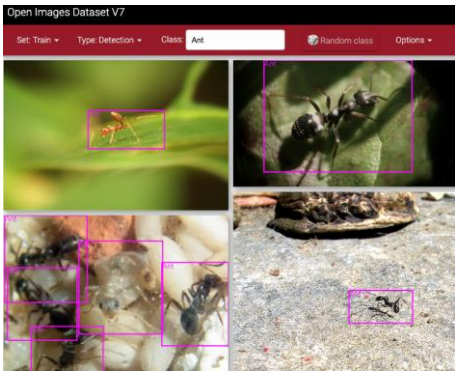


Figure 5. Annotated dataset for a specific class of objects.

Such images and annotated data are loaded from the Amazon Web Services S3 bucket by such link with defined input parameters for images, directories, and class names:

curl –location –request GET 'aws s3 cp s3://open-image-dataset/:imageDir/: image/:datasetDir/:ClassesName'

The proportion of train and validation images can also be defined, but by default, it is set as 70 train images to 30 validation images.

Additionally, the service supports manually adding annotated images using tools like Label Studio, providing flexibility for incorporating custom data in case automatically annotated data is insufficient.

4.1.2. Training service

Once the data is annotated, the model training service scales the training process to leverage available hardware resources, whether CPU or GPU. This service is designed to be versatile, supporting various operating systems, including Unix-based systems like Ubuntu, macOS, and Windows. The training service uses the input data, specified hyperparameters, loss functions, and clustering methods to produce optimized weight coefficients.

The training process also involves fine-tuning these weight coefficients to identify the most optimal features for different sizes of recognized objects. The output includes the trained model and analytical metrics such as mAP, essential for evaluating model performance.

It is important to note that the local directories obtained from the annotation step must be mounted to the Docker container through volumes. The input device should have the Compute Unified Device Architecture (CUDA) scaling system and the cuDNN library for GPU operations to enable process parallelization. CUDA access is then granted from within the Docker container. If CUDA is unavailable, the training can be executed on the CPU, though it will be significantly slower.

4.1.3. Conversion service and model quantization

After training, the conversion service processes the final model weights, optimizing them for deployment on mobile devices. This process involves quantization using affine transformations to reduce the size of the weight coefficients, making the model more efficient for mobile applications. The service also generates and records metadata, including recognition anchors and other characteristics necessary for the model's integration into the application. Finally, the model is converted into the CoreML format, specifically into the MLPackage format, using Swift 5 and Xcode tools, and is ready for deployment on iOS devices.

Specific quantization methods are recommended to further reduce the model's weight coefficients and enhance performance. The proposed affine transformation quantization method reduces precision to 8 bits. In contrast, using a lookup table formed through k-means clustering, the quantization method can reduce precision further to 4 bits.

As a result, the weights are quantized to 8-bit/4-bit precision for floating-point numbers, which reduces the model size by 2x or 4x, respectively. However, this reduction in model size leads to a linear decrease in recognition quality. Therefore, quantization may not be necessary for smaller models with fewer than three output layers when adapting the model to mobile platforms. In such cases, this step can be skipped.

Furthermore, the developed model's precision can be increased from 16-bit half-precision to 32-bit single-precision. However, this would significantly increase the hardware requirements for object recognition tasks, which are difficult to achieve with mobile platforms.

5. Results and analysis

For evaluating the results, it is essential to identify the key challenges that recognition systems encounter using predefined metrics. One of these challenges is assessing the efficiency of the recognition model based on input parameters during training. The second challenge is benchmarking the performance, which varies depending on the type of model used. In the following section, I will examine these results in detail, thoroughly analyzing the system's performance and effectiveness. The results described below were evaluated during my Ph.D. research (Kushnir, 2023).

5.1. Recognition models efficiency with k-means++ clustering

The key metrics discussed in this study include recall (**R**), precision (**P**), true positives (**TP**), false negatives (**FN**), F1-score (**F1**), model weight size (**w**), and mean Average Precision (**mAP**). Additionally, clustering methods like K-means/**K-means++** were defined, as long as the size of the training set and the input resolution of the neural network.

For this study, four main types of YOLOv4-based neural network models were trained and implemented. These models vary by several factors: the number of output layers, with some models having two layers (tiny models) and others three layers (regular models); the maximum input image resolution, set at either 512 or 416 pixels; and the clustering method used, which was either the enhanced K-means++ or the standard K-means.

To further refine the models, a Smoothing Compression Filter (SCF) was set at 0.9, and the Intersection Over Union (IoU) threshold was set at 0.2 to minimize the impact of unlikely results. The evaluation results are shown in **Table 2,** verified in my Ph.D. research (Kushnir, 2023).

Table 2. Efficiency metrics for the developed CNN models in object recognition tasks depend on the chosen clustering method, input image resolution, and number of output layers.

Metric	**R (%)**	**P (%)**	**FN**	**TP**	**F1 (%)**	**w (MB)**	**mAP (%)**
CNN Model	↑	↑	↑	↑	↑	↑	↑
Yolov4_retr_416 (k-means++)	**91.9**	97.4	**27**	**272**	**96.77**	256	**97.2**
Yolov4_retr_512 (k-means++)	**93.7**	98.2	**19**	**284**	95.32	256	**96.2**
Yolov4_416 (k-means)	91.3	**99.2**	32	267	91.87	246	94.2
Yolov4_512 (k-means)	92.2	**98.1**	18	277	**93.787**	246	93.6
Yolov4_retr_tiny_416 (k-means++)	**86.1**	90.4	41	**268**	**91.5**	24.2	**86.92**
Yolov4_retr_tiny_512 (k-means++)	**87.4**	92.5	**38**	261	**91.8**	25.2	82.99
Yolov4_tiny_416 (k-means)	81.5	**95.4**	42	256	88.6	24.3	81.21
Yolov4_tiny_512 (k-means)	82.2	**97.2**	**39**	**262**	87.2	25.1	**83.11**

The test involved four implemented CNN models and four standard CNN models of the same type for comparison. When using the K-means++ clustering algorithm during anchor generation, the generated model shows a 5% improvement in mAP.

Additionally, it is observed that precision (P) and recall (R) mutually constrain each other: as the R-value increases, the P-value decreases in a linear progression. Thus, when using the K-means++ clustering method, the R-value increases by an average of 3-4%, while the P-value decreases by 2-3%. The F1-score is used to balance the R and P values. Therefore, the higher the F1 score, the more accurate the model is overall. On average, the F1 score improved by 5-6% for most input models.

The quantitative error values of FN and TP are linear and depend on the CNN's effectiveness during testing. At the same time, the number of FN errors is slightly higher for models with two output layers (tiny models) than their counterparts with three output layers (Figure 6).

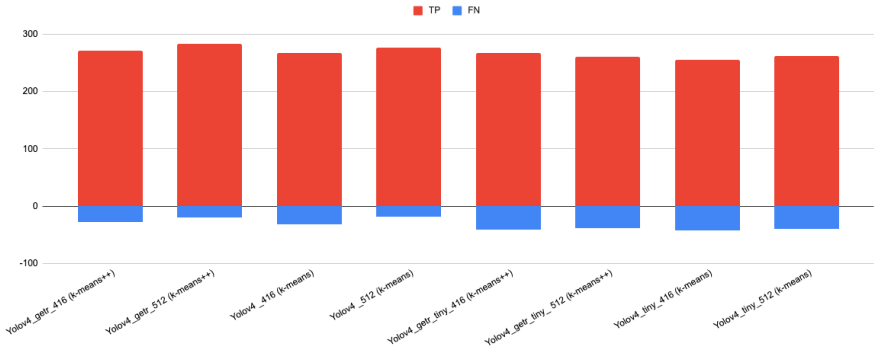


Figure 6. Comparison of FN (blue) and TP (red) values for the tested CNN models.

The difference in FN values is acceptable, as the recognition accuracy for these smaller models remains high, within the range of 85 mAP. The resolution of the input images directly influences the recognition efficiency of the CNN models. At a resolution of 512 pixels, the possible processing area for input images increases, thereby improving all efficiency characteristics of the model by 8-10%. However, the higher the resolution of the input images, the more hardware resources are required for object recognition tasks. Therefore, choosing a model with an optimal input image resolution of 416x416 pixels is advisable for mobile platforms.

5.2. Performance evaluation of the quantized CNN models on the mobile platform

The key metrics discussed in this study include **FPS**, **BFLOPs**, time to display results on the screen after processing begins ($\boldsymbol{T_{frame}}$), median time to perform an object prediction operation ($\boldsymbol{T_{predict}}$), median time to load the model onto the tested device ($\boldsymbol{T_{load}}$), model weight size (**w**), quantization level of the CNN models (**q**), hardware resources, processor types, and software tools. Additionally, **mAP** was used to evaluate the effectiveness of CNN on different mobile platforms.

To evaluate the performance of the developed CNN YOLO models on mobile platforms in real-time video imaging, it is appropriate to compare the available software and hardware resources according to the defined parameters and evaluation metrics. For testing, the following were selected:

- CNN models with two output layers (tiny models) and three output layers (regular models), with some models converted to CoreML format for iOS mobile devices;
- Input image resolution fixed at 416x416 pixels for all CNN models and devices on which they were tested;
- Hardware resources: For iOS, the test used the CPU, GPU, and NPU acceleration chips (depending on the selected floating-point computation precision). For the embedded Jetson Nano system, the CPU and GPU acceleration chip were used;
- The overall values of $\boldsymbol{T_{predict}}$ and $\boldsymbol{T_{load}}$ metrics were determined considering the use of all possible acceleration means (CPU/GPU/NPU);
- Software tools: the CoreML framework for iOS MOS and the OpenCV library for the Jetson Nano embedded system running on the Ubuntu operating system;
- Quantization levels (**q**) of the CNN model's weight coefficients for iOS MOS:
  - 16 bits: optimal for object recognition tasks using CPU and NPU;
  - 8 bits using affine transformations;
  - 4 bits using a lookup table created with the k-means clustering method;
  - 32 bits: double precision calculations for increased model performance using more extensive hardware resources, employing both CPU and GPU;
- The standard precision of 16 bits was applied for the embedded Jetson Nano system.

Table 3 and Table 4 present the performance results of the developed CNN models, verified in Ph.D. research (Kushnir, 2023)**.**

Table 3. Performance Metrics of Two-Layer CNN Models Depending on the Quantization Level and Model Type.

	Metric / **CNN Model**	**FPS** **(frames/s)**	**BFLOP's** **(billion op.)**	**w** **(MB)**	T_{frame} (s)	$T_{predict}$ (s)	T_{load} (s)	**mAP** **(%)**
	Yolov4_retr_tiny_coreml	8.4	6.454	**23.2**	0.02	**0.111**	**0.358**	86.9
q = 32	**Yolov4 _tiny_coreml**	**9**	**8.76**	24.5	0.02	0.134	0.42	87.1
	Yolov4_retr_tiny_coreml	**30.2**	7.34	**12**	0.03	**0.042**	**0.183**	82.1
	Yolov4_retr_tiny_nano	19.1	7.84	24	0.02	0.123	0.67	86.92
q = 16	Yolov4 _tiny_coreml	30.1	8.21	12.3	0.03	0.43	0.212	86.2
	Yolov4 _tiny_nano	18.0	**8.44**	24	0.02	0.234	0.69	87.21
	Yolov4_retr_tiny_coreml	**33.3**	4.22	**6.1**	0.02	**0.067**	0.434	82.1
q = 8	**Yolov4 _tiny_coreml**	33	**4.94**	7.9	0.02	0.074	**0.383**	81.7
	Yolov4_retr_tiny_coreml	32	**2.44**	**3.2**	0.03	0.08	0.46	68.1
q = 4	Yolov4 _tiny_coreml	32	2.56	3.3	0.04	0.08	0.41	61.2

The test results should be analyzed based on the metric values. As seen in Table 3

and Table 4, depending on the quantization level (**q**) increases, the **FPS** value also increases. However, the object recognition efficiency metrics, such as **mAP** and $\boldsymbol{T_{predict}}$, decrease proportionally. When quantization is reduced to 4 bits, recognition quality drops sharply.

The values of **BFLOPs**, **w,** and $\boldsymbol{T_{load}}$ decrease linearly depending on the increase in quantization level and changes in the number of NNM output layers. In most tests, the proposed CNN model shows improved results compared to its direct analogs, with an average improvement of 5-10% across most metrics.

Table 4. Performance Metrics of Three-Layer CNN Models Depending on the Quantization Level and Model Type.

	Metric	**FPS**	**BFLOP's**	**w**	$\boldsymbol{T_{frame}}$	$\boldsymbol{T_{predict}}$	$\boldsymbol{T_{load}}$	**mAP**
	CNN Model	**(frames/s)**	**(billion op.)**	**(MB)**	(s)	(s)	(s)	**(%)**
	Yolov4_retr_coreml	5.1	49.2	257	4.6	0.427	2.6	98.1
q = 32	**Yolov4 _coreml**	5.2	**52.8**	258.5	4.2	0.428	2.5	97.9
	Yolov4_retr_coreml	**6.3**	**45.1**	**129**	3.7	**0.32**	3.55	97.3
	Yolov4_retr_nano	3.2	42.1	256	**3.2**	0.39	**2.44**	97.2
q = 16	**Yolov4 _coreml**	**6.4**	**46.2**	**129.7**	3.6	**0.34**	3.63	94.8
	Yolov4 _nano	3.3	42.3	257	**3.3**	0.4	**2.37**	94.2
	Yolov4_retr_coreml	8.1	**29.1**	64.2	2.1	**0.101**	**3.21**	82.1
q = 8	Yolov4 _coreml	8.0	28.2	65.2	2.32	0.14	3.39	85.5
	Yolov4_retr_coreml	**13.7**	18.3	33.3	1.35	0.081	**2.12**	73.1
q = 4	Yolov4 _coreml	13.4	18.1	34.1	1.43	0.083	2.43	54.2
	Metric	**FPS**	**BFLOP's**	**w**	$\boldsymbol{T_{frame}}$	$\boldsymbol{T_{predict}}$	$\boldsymbol{T_{load}}$	**mAP**
	CNN Model	**(frames/s)**	**(billion op.)**	**(MB)**	(s)	(s)	(s)	**(%)**
	Yolov4_retr_coreml	5.1	49.2	257	4.6	0.427	2.6	98.1
q = 32	**Yolov4 _coreml**	5.2	**52.8**	258.5	4.2	0.428	2.5	97.9
	Yolov4_retr_coreml	**6.3**	**45.1**	**129**	3.7	**0.32**	3.55	97.3
	Yolov4_retr_nano	3.2	42.1	256	**3.2**	0.39	**2.44**	97.2
q = 16	**Yolov4 _coreml**	**6.4**	**46.2**	**129.7**	3.6	**0.34**	3.63	94.8
	Yolov4 _nano	3.3	42.3	257	**3.3**	0.4	**2.37**	94.2
	Yolov4_retr_coreml	8.1	**29.1**	64.2	2.1	**0.101**	**3.21**	82.1
q = 8	Yolov4 _coreml	8.0	28.2	65.2	2.32	0.14	3.39	85.5
	Yolov4_retr_coreml	**13.7**	18.3	33.3	1.35	0.081	**2.12**	73.1
q = 4	Yolov4 _coreml	13.4	18.1	34.1	1.43	0.083	2.43	54.2

When comparing the performance of the CNN model on iOS MOS using the iPhone 12 hardware versus the embedded Jetson Nano device (at a 16-bit quantization level), iOS MOS has a significant advantage. This advantage is achieved through the successful combination of system processors (**NPU** and **GPU**) when solving object recognition tasks. In contrast, a roughly equivalent Jeston ARM Nvidia processor cannot provide a sufficient number of **BFLOPs**.

To verify this hypothesis regarding the use of hardware resources, a test was conducted on iOS mobile device using limited hardware resources for the $T_{predict}$ and T_{load} metrics (Figure 7 and Figure 7).

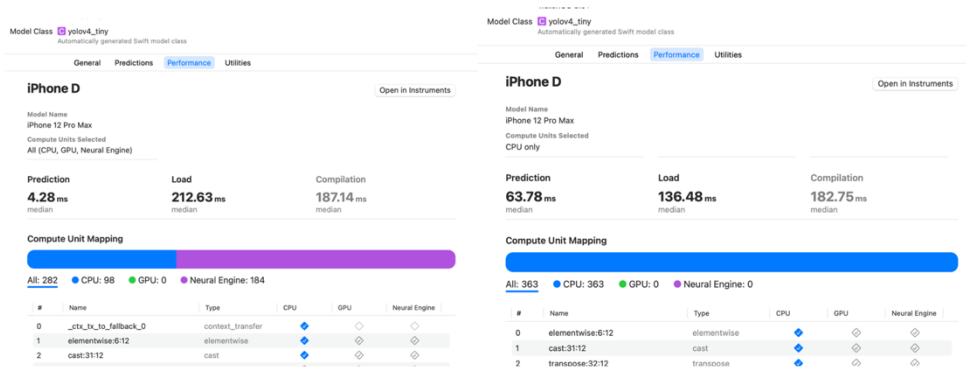


Figure 7. Usage of CPU and NPU (left) vs. CPU only (right) during testing the two-layer YOLO CNN model.

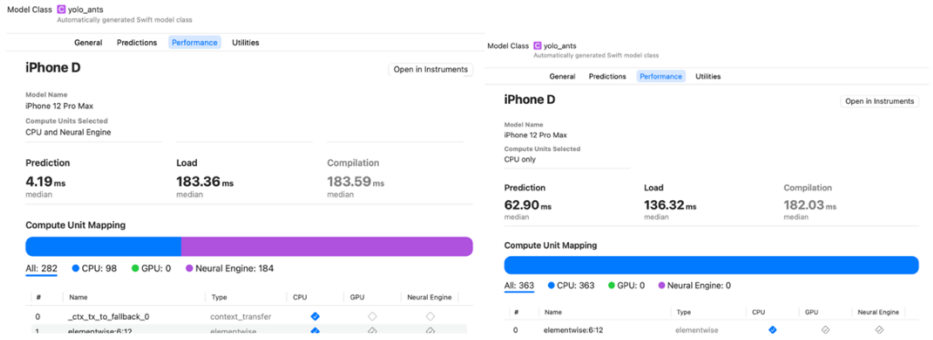


Figure 8. Usage of CPU and NPU (left) vs. CPU only (right) during testing the two-layer RETR YOLO CNN model.

The results indicate that using only the **CPU** for object recognition tasks on mobile platforms reduces the prediction (recognition) and data processing performance by 12 times. Additionally, the model loading time increases by 64%, which does not significantly impact development efficiency.

Meanwhile, using 32-bit quantization activates the **GPU** acceleration chip instead of the **NPU**, reducing the model's performance by 70%. Thus, the performance of the CNN model on iOS, when using both **CPU** and **GPU**, is approximately equivalent to the metrics of the embedded Jetson Nano OS (with a quantization level of 16 bits, $T_{predict}$ is 0.111 for iOS and 0.123 for the embedded Jetson Nano system).

The **FPS** for most tiny models exceeds the threshold value of 24 frames per second, making the developed models suitable for real-time tasks.

5.3. Model evaluation in the indoor environment with ants

The proposed methodology and frameworks were tested on an indoor ant colony of Camponotus fellah ants as they moved through the nest from one entrance to another (Figure 9). The ex situ recording setup featured a static tripod for stability, a macro lens (Apexel APL-HB2XT) with a 60mm focal length for capturing detailed images of the ants, and LED lighting to ensure optimal visibility of their movements.

Figure 9. Recognition and tracking of Camponotus fellah ants during movement. Each recognized ant is assigned a unique identifier (e.g., recognized rectangle #311 for static ant), allowing for real-time tracking of their movements using a specific color.

For this research, images and video footage were captured using the rear wide camera of an iPhone 12. The device's 12-megapixel resolution optical image stabilization camera can also handle 30FPS for CNN processing. While the camera performed well in controlled settings, challenges like ants on vertical surfaces or blurred frames affected recognition performance. Expanding the dataset to include images under these conditions could improve model robustness.

The numbers assigned near each ant indicate a unique identifier generated by the V-IOU algorithm during tracking. As previously noted, to minimize overlap between recognized bounding boxes, the SCF was set to 0.9, and the IOU threshold was configured at 0.2. These parameters ensured precise object tracking while reducing the likelihood of redundant or overlapping detections.

The injected V-IOU algorithm was applied using the specified JavaScriptCore methods for tracking. The path is defined by displaying the centroids of each unique object in distinct colors, with the option to retain this information for a specified duration. Each unique object within a particular class is identified by its color.

This information can then be applied to an analytics system for counting recognized

and tracked objects, like ants crossing specific entrances in the nest by crossing a boundary line. By leveraging the tracking algorithm's ability to distinguish individual objects, the system could provide detailed insights into movement patterns and colony dynamics.

6. Discussion

The analysis results indicated that using the k-means++ clustering method significantly improved object recognition efficiency, with an increase of 5% in mAP, 5-6% in F1, and 3-4% in R. Considering the available hardware capabilities, a model with two output layers and a resolution of 416x416 pixels was determined to be optimal for the NNM model's performance on mobile platforms.

Performance metrics analysis of the developed CNN model suggested that the optimal quantization level is 16 bits for the 2-layer model and 8 bits for the 3-layer model. The FPS for most models remained around 24 FPS, which is sufficient for real-time tasks.

Models on the iOS platform, converted to CoreML format, demonstrated the highest performance, as this mobile operating system effectively leverages the system's processors (NNA and GPU) for object recognition tasks. In contrast, the Jeston ARM Nvidia processor could not provide sufficient BFLOPS despite having similar characteristics. The research revealed that using NPU and CPU chips increased prediction (recognition) and data processing performance ($T_{predict}$) 12 times compared to using only the CPU on iOS mobile devices.

In most performance tests, the proposed CNN RETR YOLO model showed improved results compared to its direct analogs, achieving an average of 5-10% improvement across most metrics.

Overall, this CNN model for mobile devices can effectively recognize and track small, fast-moving objects in real time.

Future work may explore migrating the tracking algorithm to a native Swift implementation to further optimize performance and fully leverage hardware-specific accelerations available on iOS devices.

The system's compatibility with any iOS device offers significant potential for scalable and accessible monitoring solutions for possible in situ applications. Future deployment in natural habitats could involve additional hardware, such as a stabilizer and retainer for consistent image capture and a portable power source to support extended operation. These enhancements, combined with the framework's portability and ease of integration, suggest that it could be effectively adapted for efficient field applications.

7. Conclusions

The study demonstrated that optimizing the CNN model with the k-means++ clustering method and specific quantization levels significantly improves object recognition and tracking on mobile platforms. Converting models to CoreML for iOS proved remarkably effective, leveraging NPU and GPU chips to enhance performance. The research also successfully applied these methods to the recognition and tracking

of ants, illustrating the model's capability to handle small, dynamic objects in real time. These findings contribute to developing advanced mobile applications for recognizing and tracking small, fast-moving objects, paving the way for further advancements in mobile-based recognition systems.

Acknowledgments

Ukraine's Ministry of Education and Science supported part of the study through the "Intelligent Design Methods and Tools for the Modular Autonomous Cyber-Physical Systems" project (registration #0119U100609).

References

Arthur, D., Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics. https://doi.org/10.5555/1283383.1283494

Bochinski, E., Senst, T., Sikora, T. (2018). Extending IOU based multi-object tracking by visual information. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). Auckland, New Zealand. https://doi.org/10.1109/AVSS.2018.8639144

Bochkovskiy, A., Redmon, J., Sinigardi, S., Hager, T., Jaled M.C. et al. (2021). AlexeyAB/darknet (version yolov4). Zenodo. https://doi.org/10.5281/zenodo.562267

Gudauskas, J., Petkutė, G., Trakšelis, K., Kriščiūnas, A. (2024). UAV-based traffic intensity analysis framework: A case study on pedestrian crossings. *Baltic Journal of Modern Computing*, **12**(1), 102-115. https://doi.org/10.22364/bjmc.2024.12.1.07

Kushnir, D. (2022). Methods and means for small dynamic objects recognition and tracking. Computers, Materials & Continua, **73(**1), 1933-1949. https://doi.org/10.32604/cmc.2022.030016

Kushnir, D. (2022). Ants dataset (indoor/outdoor Messor Structor) + trained YOLOv4 weights. Mendeley Data. https://doi.org/10.17632/zprk7wkf9j.1

Kushnir, D. (2023). Methods and means of searching and recognizing objects in video images on the mobile platform in real-time, PhD thesis, Lviv Polytechnic National University, Lviv, Ukraine. https://lpnu.ua/sites/default/files/2023/radaphd/23565/diskushnir.pdf

Li, R., Wang, Y., Liang, F., Qin, H., Yan, J., Fan, R. (2019). Fully quantized network for object detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2810-2819).

Marques, O. (2020). Machine learning with Core ML. In *Image Processing and Computer Vision in iOS* (pp. 29-40). Springer. https://doi.org/10.1007/978-3-030-54032-6_4

Novák, M. (2020). Secure JavaScript UI rendering for iOS using Swift.

Popp, S., Dornhaus, A. (2024). Collective search in ants: Movement determines footprints, and footprints influence movement. *PLOS ONE*, **19**(4), e0299432. https://doi.org/10.1371/journal.pone.0299432

Redmon, J., Divvala, S. K., Girshick, R. B., Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, 779 - 788.

Smith, R., Patel, K. (2023). IntelliBeeHive: Real-time monitoring of honeybee activity using machine learning. arXiv Preprint. https://doi.org/10.48550/arXiv.2309.08955

Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N. (2020). Label Studio: Data labeling software. https://github.com/heartexlabs/label-studio

Ulrich, J., Arvin, F., Rojas, N. et al. (2024). Autonomous tracking of honey bee behaviors over long-term periods with cooperating robots. *Science Robotics*, **9**(47), eadn6848. https://doi.org/10.1126/scirobotics.adn6848

Wang, Y., Guan, Y., Liu, H., Jin, L., Li, X., Guo, B., Zhang, Z. (2023). VV-YOLO: A vehicle view object detection model based on improved YOLOv4. *Sensors*, **23**(7), 3385. https://doi.org/10.3390/s23073385

Received August 14, 2024, revised January 23, 2025, accepted January 25, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 96–127
https://doi.org/10.22364/bjmc.2025.13.1.06

Advancing Equal Opportunity Fairness and Group Robustness through Group-Level Cost-Sensitive Deep Learning

Modar SULAIMAN, Nesma Talaat Abbas MAHMOUD, Kallol ROY

University of Tartu, Institute of Computer Science, Tartu, Estonia

modar.sulaiman@ut.ee, nesma.mahmoud@ut.ee, kallol.roy@ut.ee

ORCID: 0000-0002-7322-078X, ORCID: 0000-0002-5618-2953,
ORCID: 0000-0002-6557-2689

Abstract. Machine learning systems deployed in high-stakes decision-making scenarios increasingly face challenges related to fairness, spurious correlations, and group robustness. These systems can perpetuate or amplify societal biases, particularly affecting protected groups defined by sensitive attributes such as race or age. This paper introduces a novel cost-sensitive deep learning approach at different group levels that simultaneously addresses these interconnected challenges. Thus, our research uncovers a fundamental synergy between group robustness and group fairness. By developing a technique that enhances group fairness, we also improve the model's group robustness to spurious correlations. This approach encourages the model to focus on causally relevant features rather than misleading associations. We propose a comprehensive methodology that specifically targets group-level class imbalances, a crucial yet often overlooked aspect of model bias. By incorporating different misclassification costs at the group level, our approach, Group-Level Cost-Sensitive Learning (GLCS), provides a principled framework for handling both dataset-wide and group-specific class imbalances using different constraints in an optimization framework. Through targeted interventions for underrepresented subgroups, we demonstrate simultaneous improvements in equal opportunity fairness and worst-group performance, ensuring similar true positive rates across demographic groups while strengthening overall group robustness. Extensive empirical evaluation across diverse datasets (CelebA, UTKFace, and CivilComments-WILDS) demonstrates that our method effectively mitigates performance disparities and promotes more equitable outcomes without sacrificing overall model accuracy. These findings present evidence that addressing fundamental data distribution issues at the group level can naturally lead to fairer and more robust machine learning systems. Our work has significant implications for the ethical deployment of machine learning in critical domains such as healthcare, finance, and criminal justice, offering a practical path toward more equitable and reliable automated decision-making systems.

Keywords: Fairness in Machine Learning, Group Robustness, Cost-Sensitive Deep Learning, Bias.

1 Introduction

Machine learning systems have become increasingly prevalent in high-stakes decision-making scenarios, from lending and hiring to healthcare and criminal justice (Barocas et al., 2023; Chouldechova and Roth, 2020). However, these automated systems can perpetuate or amplify existing societal biases, leading to discriminatory outcomes against protected groups defined by sensitive attributes such as race or gender (Mehrabi et al., 2021). This has prompted extensive research in fair machine learning, which aims to develop algorithms that maintain high predictive performance while ensuring equitable treatment across different demographic groups (Du et al., 2020). Various fairness metrics and mitigation strategies have emerged, including statistical parity (Dwork et al., 2012), equal opportunity (Hardt et al., 2016), and individual fairness (Dwork et al., 2012). These approaches fall into three categories: pre-processing techniques that modify training data, in-processing methods that incorporate fairness constraints during model training, and post-processing approaches that adjust model outputs (Caton and Haas, 2024). Despite these advances, achieving fairness while maintaining model performance remains challenging, due to multiple competing criteria (Kleinberg et al., 2016). Moreover, the context-dependent nature of fairness requires careful consideration of domain-specific requirements and societal implications when designing and deploying fair ML systems (Selbst et al., 2019).

A fundamental challenge in achieving fairness lies in the presence of spurious correlations, where machine learning models inadvertently learn misleading associations between features and outcomes that do not reflect true causal relationships. These correlations arise from various sources, including sampling biases, historical data imbalances, or coincidental patterns in training datasets. As a result, the impact of spurious correlations extends beyond mere performance issues, as models that rely on spurious features can inadvertently perpetuate or amplify existing societal biases, leading to discriminatory outcomes that disproportionately affect minority groups. The concept of group robustness (which involves training models to achieve strong performance across all predefined groups within the dataset), measured by the minimum accuracy across all groups (worst-accuracy), is also not immune to these challenges. Models trained with standard empirical risk minimization (ERM) often exhibit poor performance on underrepresented groups due to both geometric and statistical skew on the input training data.

This paper investigates the fundamental synergy between group robustness and group fairness in machine learning models. We demonstrate that our approach, designed to enhance the group fairness metric, also boosts group robustness by ensuring consistent performance across all subgroups. This addresses the common problem of models exploiting spurious patterns that unfairly impact minority groups. We propose a novel Group-Level Cost-Sensitive Framework (GLCS) to address these challenges. Our proposed GLCS framework leverages cost-sensitive deep learning (Khan et al., 2017; Zhou and Liu, 2005) and addresses class imbalance challenges by explicitly incorporating misclassification costs at the group level into the learning process. Our proposed methodology differs fundamentally from conventional techniques of random oversampling, undersampling, or synthetic minority oversampling (SMOTE) by modifying the underlying learning objective rather than manipulating the dataset distribution.

The main contributions of the paper are laying the foundations at the intersection of cost-sensitive deep learning, group fairness, and group robustness in machine learning. The key contributions are outlined as follows:

1. **Novel Group-Level Cost-Sensitive Framework (GLCS):** We introduce a pioneering cost-sensitive deep learning framework that addresses group-level class imbalances, enabling more nuanced handling of demographic disparities in machine learning systems.
2. **Enhanced Fairness and Robust Performance Mechanisms:** The proposed novel cost-sensitive optimization technique GLCS mitigates performance disparities on diverse datasets (CelebA, UTKFace, and CivilComments-WILDS) by strategically balancing group-level representations, thereby improving equal opportunity fairness without compromising overall model accuracy.
3. **Comprehensive Empirical Validation:** Our extensive experimental results validate the generalizability and effectiveness of our approach, showcasing consistent improvements in both group robustness and group fairness, with a particular focus on equal opportunity.

The paper is structured as follows: Section 2 presents a comprehensive review of related work in group fairness, group robustness, fairness and class imbalance, threshold optimization, and cost-sensitive learning. Section 3 establishes the necessary preliminaries and theoretical foundations. Section 4 introduces our Group-Level Cost-Sensitive (GLCS) framework, detailing its mathematical formulation and implementation. Section 5 describes the datasets and baselines employed in our experimental evaluation, while Section 6 outlines our evaluation metrics. Section 7 details the experimental setup and implementation details. Finally, Section 8 presents our results and discusses their implications for group fairness and group robustness in machine learning.

2 Related Works:

Fairness in Machine Learning. The debiasing techniques strategically categorized into three primary approaches: pre-processing, in-processing, and post-processing methods (Wan et al., 2023). While pre-processing and post-processing techniques offer pragmatic interventions, our research specifically focuses on in-processing debiasing methods, which have garnered substantial scholarly attention for their sophisticated approach of embedding fairness constraints directly. These intrinsic fairness techniques, pioneered by seminal works from (Dwork et al., 2012; Hashimoto et al., 2018; Kearns et al., 2018), represent a paradigm shift towards algorithmically engineered fairness. (Zafar et al., 2019) studies constrained optimization by incorporating fairness measures as regularisation terms or constraints.

Cost-Sensitive Learning. Cost-sensitive learning adaptively weighs the importance of different classes during the training process. This is typically achieved through the modification of the loss function in neural networks. The approach is effective in real-world applications of medical diagnosis, fraud detection, or rare event prediction, where misclassification costs are inherently asymmetric. Recent developments in this area have

introduced several innovative methodologies. (Zhou and Zhang, 2016) employed cost-sensitive learning to mitigate the problem of misclassifications of minority or critical classes. The class-balanced loss function (Cui et al., 2019) addresses the long-tailed distribution problem by introducing a weighting factor that is inversely proportional to the effective number of samples. Margin-based approaches (Cao et al., 2019) focus on enhancing the decision boundary's quality by incorporating cost-sensitivity into the margin requirements. Additionally, (Sangalli et al., 2021) uses constrained optimization to train neural networks to improve neural network performance on critical and under-represented classes.

Fairness and Class Imbalance. The intricate relationship between fairness and class imbalance has emerged as a critical research domain in machine learning, with scholars developing sophisticated methodologies to address simultaneous challenges of bias mitigation and distributional disparities. (Dablain et al., 2022) introduced Fair Over-Sampling (FOS), a pioneering approach that simultaneously addresses class imbalance and protected feature bias by generating synthetic minority class instances while encouraging classifiers to minimize reliance on sensitive attributes. Complementing this work, (Hirzel and Ram, n.d.) developed Orbis, an adaptable oversampling algorithm capable of fine-tuned optimization across fairness and accuracy dimensions. (Yan et al., 2020) critically demonstrated how conventional balancing techniques can inadvertently exacerbate unfairness, introducing a novel fair class balancing method that enhances model fairness without explicit sensitive attribute manipulation. (Tarzanagh et al., 2023) advanced this discourse through a tri-level optimization framework incorporating local, fair, and class-balanced predictors, theoretically demonstrating improved classification and fairness generalization. (Subramanian et al., 2021) further expanded these investigations by evaluating long-tail learning methods across sentiment and occupation classification domains, empirically validating fairness enforcement techniques' effectiveness in mitigating demographic biases and class imbalance. (Shui et al., 2022) contributed a principled bilevel objective approach, demonstrating an innovative method for developing fair predictors that simultaneously manage group sufficiency and generalization error.

Group Robustness. Recent machine learning research has developed sophisticated methods to address the problem of group robustness. (Sagawa et al., 2019) introduced Group Distributionally Robust Optimization (Group-DRO), which optimizes a soft version of the worst-group loss. (Liu et al., 2021) proposed Just Train Twice (JTT), a method that employs a two-stage training strategy: Initially, a standard ERM model is trained for several epochs. In the subsequent stage, a refined model is trained by upweighting the training examples that the initial ERM model misclassified. Complementing these approaches, Kirichenko et al.(Kirichenko et al., 2022) demonstrated through Deep Feature Reweighting (DFR) that simple last layer retraining can match or surpass state-of-the-art methods on spurious correlation benchmarks with significantly reduced computational complexity. Building upon this insight, (Qiu et al., 2023) developed Automatic Feature Reweighting (AFR), which retrains the last layer of ERM-trained model with a weighted loss that upweights minority group examples by emphasizing instances where the ERM model performs poorly.

Threshold Optimization. Threshold optimization in classification represents a sophisticated computational domain, with seminal works (Lipton et al., 2014; Koyejo et al., 2014; Sanchez, 2016) systematically exploring methodological approaches for determining optimal decision boundaries. Research has advanced through receiver operating characteristic (ROC) curve analysis for identifying optimal operating points (Freeman and Moisen, 2008), cost-sensitive threshold adjustment techniques that explicitly incorporate domain-specific loss functions and contextual constraints into the threshold selection process (Robles et al., 2020), unified theoretical frameworks (Hernández-Orallo et al., 2012) offering comprehensive computational strategies for threshold optimization that transcend traditional binary classification paradigms, and probabilistic methodologies for adaptive threshold determination which significantly enhancing the precision and reliability of predictive models across diverse computational domains (Kazemi et al., 2023), thereby providing a comprehensive approach to optimizing classification thresholds with nuanced consideration of performance, constraints, and contextual requirements. For a more detailed explanation of the threshold optimization method employed in our experiment, please refer to Section 7.3.

3 Mathematical Preliminaries

Definition 1 (Group Fairness). Group fairness is a fundamental concept in machine learning and algorithmic decision-making, particularly relevant when the outcomes affect individuals from different demographic or social groups. The aim of group fairness is to ensure that model predictions are equitable across groups defined by sensitive attributes such as gender, race, or religion.

Definition 2 (Equal Opportunity). It requires that a model achieves the same true positive rate (TPR) for different subgroups when considering only instances with a positive label (Hardt et al., 2016). Formally, it is defined as:

$$P(\hat{Y} = 1|S = 0, Y = 1) = P(\hat{Y} = 1|S = 1, Y = 1),$$

where $\hat{Y}$ represents the predicted outcome, S is the sensitive attribute, and Y is the true label. This condition ensures that individuals from different groups who are actually positive (i.e., have a positive true label) have an equal probability of being classified as positive by the model.

Definition 3 (Equalized Odds). It extends the concept of Equal Opportunity by requiring that both the true positive rate (TPR) and the false positive rate (FPR) be equal across different groups (Hardt et al., 2016). It can be expressed as:

$$P(\hat{Y} = y|S = 0, Y = y) = P(\hat{Y} = y|S = 1, Y = y), \quad y \in \{0, 1\}.$$

This metric ensures that the model's performance is consistent across groups in terms of both correctly identifying positives and avoiding false positives. By examining these metrics, we aim to provide a thorough evaluation of fairness in our models, ensuring they treat all groups equitably.

Definition 4 (Group Robustness). It focuses on maintaining consistent and fair model performance across all subgroups (Liu et al., 2021; LaBonte et al., 2024). Group robustness optimizes strategies to focus on: (i) Identifying and mitigating spurious correlations. (ii) Optimizing performance for worst-performing groups. (iii) Maintaining high overall accuracy while improving minority group performance.

Definition 5 (Augmented Lagrangian Method (ALM)). The Augmented Lagrangian Method (ALM), also known as the method of multipliers, is a powerful optimization technique that bridges the gap between constrained and unconstrained optimization problems. Introduced by Bertsekas (1976).

Definition 6 (Class-based Partitioning). The dataset M can be partitioned into positive (critical) and negative classes as follows:

$$\begin{aligned} P &= \{x_i^p\}_{i=1}^{|P|} \quad \text{(positive class samples)} \\ N &= \{x_i^n\}_{i=1}^{|N|} \quad \text{(negative class samples)} \end{aligned} \tag{1}$$

where $|P| < |N|$, indicating P represents the minority class (Sangalli et al., 2021).

Definition 7 (Protected Attribute-based Partitioning). Let $s \in \{0, 1\}$ denote the protected attribute (e.g., gender or race). We partition the dataset M into two disjoint subsets based on this attribute:

$$\begin{aligned} Z_1 &= \{x_i^{s_1}\}_{i=1}^{|Z_1|} \quad \text{(group with } s = 1\text{)} \\ Z_0 &= \{x_i^{s_0}\}_{i=1}^{|Z_0|} \quad \text{(group with } s = 0\text{)} \end{aligned} \tag{2}$$

where in our discription we select $|Z_1| < |Z_0|$, establishing Z_0 as the majority group (non-protected group) and Z_1 is the minority group (protected group).

The further partitioning of each protected attribute group (Intersectional Subgroups) based on the true labels $y \in \{0, 1\}$ gives the following definitions.

Definition 8 (Protected Group Partitioning (Z_1)).

$$\begin{aligned} Z_{1,1} &= \{x_i^{s_1,y_1}\}_{i=1}^{|Z_{1,1}|} \quad \text{(positive class, } y = 1\text{)} \\ Z_{1,0} &= \{x_i^{s_1,y_0}\}_{i=1}^{|Z_{1,0}|} \quad \text{(negative class, } y = 0\text{)} \end{aligned}$$

where $|Z_{1,1}| < |Z_{1,0}|$, indicating $Z_{1,1}$ is the minority class within Z_1.

Definition 9 (Non-Protected Group Partitioning (Z_0)).

$$\begin{aligned} Z_{0,1} &= \{x_i^{s_0,y_1}\}_{i=1}^{|Z_{0,1}|} \quad \text{(positive class, } y = 1\text{)} \\ Z_{0,0} &= \{x_i^{s_0,y_0}\}_{i=1}^{|Z_{0,0}|} \quad \text{(negative class, } y = 0\text{)} \end{aligned}$$

where $|Z_{0,1}| < |Z_{0,0}|$, indicating $Z_{0,1}$ is the minority class within Z_0.

Group and Subgroup Size Relationship In our theoretical framework, while we initially establish the notational convention that $|Z_1| < |Z_0|$, $|Z_{1,1}| < |Z_{1,0}|$, and $|Z_{0,1}| < |Z_{0,0}|$, we acknowledge that group and subgroup size relationships can exhibit significant variation across different experimental contexts and datasets. Specifically, the relative size constraints may be inverted in certain scenarios, such that $|Z_{1,0}| < |Z_{1,1}|$, and/or $|Z_{0,0}| < |Z_{0,1}|$. Therefore, a critical preliminary step when applying the Group-Level Cost-Sensitive Deep Learning (GLCS) framework is to rigorously characterize and distinguish between minority and majority groups/subgroups to ensure appropriate methodological implementation.

4 Proposed Method

In this paper, building upon the work of (Sangalli et al., 2021), we propose an innovative method (GLCS) formulated as a constrained optimization system for achieving equal opportunity in classification. While (Sangalli et al., 2021) focused on imbalanced dataset classification using constraints (3a) and (3b), we extend their framework by introducing additional constraints (3c) and (3d) to explicitly enforce equal opportunity across protected groups expressed as:

$$\min_{\theta} F(\theta) \quad \text{subject to:} \tag{3a}$$

$$\sum_{k=1}^{|N|} \max\Big(0, -(f_\theta(x_j^p) - f_\theta(x_k^n)) + \delta\Big) = 0, \quad \forall j \in \{1, \ldots, |P|\} \tag{3b}$$

$$\sum_{k=1}^{|Z_{1,0}|} \max\Big(0, -(f_\theta(x_l^{s_1,y_1}) - f_\theta(x_k^{s_1,y_0})) + \delta\Big) = 0, \quad \forall l \in \{1, \ldots, |Z_{1,1}|\} \tag{3c}$$

$$\sum_{k=1}^{|Z_{0,0}|} \max\Big(0, -(f_\theta(x_r^{s_0,y_1}) - f_\theta(x_k^{s_0,y_0})) + \delta\Big) = 0, \quad \forall r \in \{1, \ldots, |Z_{0,1}|\} \tag{3d}$$

where: $f_\theta(\cdot) : \mathcal{X} \to [0, 1]$ is the DNN's output probability function, θ represents the DNN parameters and $\delta > 0$ is the margin parameter. The above constraints enforces three levels of discrimination prevention by (i) ensuring separation between positive and negative classes (ii) enforcing class separation within the protected and non-ptotected groups (iii) optimizing overall AUC performance. This hierarchical constraint system simultaneously addresses both class imbalance and equal opportunity objectives, ensuring consistent performance across all subgroups while maintaining strong overall classification performance. Subsequently, we derive an equivalent unconstrained form of the above constrained system defined in equations (3a)-(3d) using the augmented Lagrangian method (ALM):

$$\mathcal{L}_\mu(\theta,\lambda) = F(\theta) + \underbrace{\frac{\mu_1}{2|P||N|}\sum_{j=1}^{|P|} q_j^2 + \frac{1}{|P||N|}\sum_{j=1}^{|P|}\lambda_j q_j}_{\text{Global Class Separation}} + \underbrace{\frac{\mu_2}{2|Z_{1,1}||Z_{1,0}|}\sum_{l=1}^{|Z_{1,1}|} q_l^2 + \frac{1}{|Z_{1,1}||Z_{1,0}|}\sum_{l=1}^{|Z_{1,1}|}\lambda_l q_l}_{\text{Class Separation Within Protected Group}} + \underbrace{\frac{\mu_3}{2|Z_{0,1}||Z_{0,0}|}\sum_{r=1}^{|Z_{0,1}|} q_r^2 + \frac{1}{|Z_{0,1}||Z_{0,0}|}\sum_{r=1}^{|Z_{0,1}|}\lambda_r q_r}_{\text{Class Separation Within Non-Protected Group}} \tag{4}$$

where the constraint violations q are defined as:

$$q_j = \sum_{k=1}^{|N|} \max\left(0, -(f_\theta(x_j^p) - f_\theta(x_k^n)) + \delta\right) \qquad \text{(global)}$$

$$q_l = \sum_{k=1}^{|Z_{1,0}|} \max\left(0, -(f_\theta(x_l^{s_1,y_1}) - f_\theta(x_k^{s_1,y_0})) + \delta\right) \qquad \text{(protected)}$$

$$q_r = \sum_{k=1}^{|Z_{0,0}|} \max\left(0, -(f_\theta(x_r^{s_0,y_1}) - f_\theta(x_k^{s_0,y_0})) + \delta\right) \qquad \text{(non-protected)}$$

Here, $\mu_1, \mu_2, \mu_3 > 0$ are penalty coefficients for quadratic terms; $\lambda_j, \lambda_l, \lambda_r$ are Lagrange multipliers for positive samples in respective groups and $\delta > 0$ is the margin parameter. This unconstrained formulation facilitates (i) asymmetric treatment by different handling of positive and negative classes in M, Z_1, and Z_0, reflecting their relative importance, (ii) performance focus by prioritizing reduction of False Positive Rate (FPR) at high True Positive Rate (TPR) regions for all groups.

5 Datasets and Baselines

Our empirical evaluation leverages three widely-recognized datasets in fairness-aware machine learning research: CelebA (Liu et al., 2015) and UTKFace (Zhang et al., 2017) for facial attribute analysis and demographic fairness assessment, and CivilComments-WILDS (Koh et al., 2021) for evaluating group robustness under distribution shifts. These datasets were selected for their diverse data modalities and comprehensive demographic annotations, enabling rigorous evaluation of both algorithmic fairness and group robustness. The facial analysis datasets present unique challenges through their demographic distributions and attribute correlations, while CivilComments-WILDS offers extensive toxic comment classifications across varied demographic groups. This diverse dataset selection facilitates thorough validation of our proposed technique across multiple domains and fairness criteria.

5.1 CelebA Dataset

The CelebFaces Attributes (CelebA) dataset (Liu et al., 2015) comprises 202,599 celebrity images with 40 binary attribute annotations, establishing itself as a benchmark dataset in fairness-aware machine learning research (Han et al., 2024). Following the Fair Fairness Benchmark (FFB) preprocessing protocol (Han et al., 2024), we focus on the binary classification task of "Wavy Hair" (y) prediction with "Gender" (s) as the protected attribute. The dataset is split into training (80%, 162,770 samples), validation (10%, 19,867 samples), and test sets (10%, 19,962 samples). Table 1 presents the distribution statistics across gender groups ($s = 1$ for male, $s = 0$ for female) and target attributes ($y = 1$ for wavy hair presence), providing crucial insights into potential data distribution biases.

Target Attribute Distribution (Wavy Hair)		
Positive Class ($y = 1$)		51,982
Negative Class ($y = 0$)		110,788
Protected Attribute Distribution (Gender)		
Male ($s = 1$)		68,261
Female ($s = 0$)		94,509
Intersectional Distribution		
Male with Wavy Hair	$P(s = 1 \mid y = 1)$	9,762
Male without Wavy Hair	$P(s = 1 \mid y = 0)$	58,499
Female with Wavy Hair	$P(s = 0 \mid y = 1)$	42,220
Female without Wavy Hair	$P(s = 0 \mid y = 0)$	52,289

Table 1. CelebA Dataset Statistics and Demographic Distribution

5.2 UTKFace Dataset

The UTKFace dataset (Zhang et al., 2017) contains over 20,000 facial images annotated with age, gender, and ethnicity attributes, making it particularly suitable for investigating intersectional fairness in facial analysis tasks. The dataset exhibits balanced distributions across major demographic factors, with 12,661 young and 11,044 old subjects, and near-equal gender representation (12,391 male, 11,314 female). Table 2 reveals notable age-gender interactions, with males showing higher representation in older age groups (6,854 vs. 5,537) and females in younger groups (7,124 vs. 4,190). Following (Han et al., 2024), images were standardized to 48x48 pixels with 3 color channels and partitioned into training (18,964 samples), validation (2,371 samples), and test sets (2,370 samples), enabling robust evaluation of algorithmic fairness across demographic intersections.

Demographic	Age Group		Total
Group	Old (y = 1)	Young (y = 0)	Sample
Male	6,854	5,537	12,391
Female	4,190	7,124	11,314
Total	11,044	12,661	23,705

Table 2. Demographic Distribution of Age Categories in UTKFace Dataset

Split	Number of Comments	Distribution (%)
Training	269,038	59.79
Validation	45,180	10.04
Test	133,782	29.73
Total	450,000	100.00

Table 3. Data Distribution in CivilComments-WILDS Dataset

5.3 CivilComments-WILDS Dataset

The CivilComments-WILDS dataset (Koh et al., 2021), derived from (Borkan, Dixon, Sorensen, Thain and Vasserman, 2019), contains 450,000 online comments annotated for toxicity and eight demographic identity mentions (gender (male, female), sexual orientation (LGBTQ), race (black, white), and religion (Christian, Muslim, or other)). This dataset is particularly valuable for studying group robustness due to potential spurious correlations between demographic mentions and toxicity labels. Following (Koh et al., 2021), we define 16 overlapping groups—(a, toxic) and (a, non-toxic) for each demographic identity a. Table 3 illustrates the distribution of comments across the dataset splits. Our analysis using the Empirical Risk Minimization (ERM) approach identified comments mentioning Christian identity as the worst-performing group, which we subsequently designated as the sensitive attribute in our GLCS framework. The effectiveness of our proposed GLCS method is assessed using worst-group accuracy metrics, detailed in Section 6, with implementation specifics discussed in Sections 5.5 and 7.5.

5.4 Baselines for group Fairness with CelebA and UTKFace Datasets

We compare our proposed GLCS method against the following baselines for group fairness on CelebA and UTKFace datasets: Empirical Risk Minimization (ERM) (Vapnik, 1991) and DiffEopp (Chuang and Mroueh, 2021; Hardt et al., 2016). ERM is a foundational machine learning technique that aims to minimize the empirical risk on the training dataset (Vapnik, 1991). ERM focuses on optimizing the performance of the model on the observed data, often without considering fairness constraints. On the other hand, DiffEopp is a gap regularization method to address the equal opportunity criterion. DiffEopp ensures that the true positive rates are equal across different demographic groups. In this paper, we utilize the implementation of DiffEopp as provided in the Fair Fairness Benchmark (FFB) (Han et al., 2024). These baselines, ERM and DiffEopp, were selected primarily to thoroughly assess the performance of our GLCS

framework in terms of both predictive accuracy and equal opportunity fairness. This comprehensive evaluation allows us to clearly demonstrate the advantages and trade-offs of our approach.

5.5 Baseline Methods for Group Robustness in CivilComments-WILDS Dataset

In our comprehensive investigation of group robustness, we evaluate several state-of-the-art methods for mitigating performance disparities across demographic groups. Our baseline approaches encompass a range of sophisticated techniques: (ERM), Just Train Twice (JTT) (Liu et al., 2021); Deep Feature Reweighting (DFR) (Kirichenko et al., 2022); Automatic Feature Reweighting (AFR) (Qiu et al., 2023); and Group Distributionally Robust Optimization (Group-DRO) (Sagawa et al., 2019).

6 Metrics

In this section, we discuss different metrics used in our experiments to validate the efficacy of our proposed GLCS approach. The metrics are classified mainly into (i) Threshold-Agnostic Performance Metrics (ii) Threshold-Dependent Performance Metrics (iii) Group Fairness Metrics (iv) Nuanced Metrics (v) Group Robustness Metric.

Threshold-Agnostic Performance Metrics. In the evaluation of binary classification models, several threshold-agnostic performance metrics provide comprehensive insights into machine learning models behavior and efficacy. *Precision-Recall Area Under the Curve (PR-AUC)*, *Receiver Operating Characteristic Area Under the Curve (ROC-AUC)*, *Brier score*, and *AUC-PR Gain* are such pivotal metrics. PR-AUC is important in scenarios with class imbalance, as it focuses on the positive class and is less affected by a large number of true negatives. ROC-AUC measures the model's ability to discriminate between classes by plotting the true positive rate against the false positive rate at various threshold settings. Brier Score measures the mean squared difference between the predicted probability and the actual outcome. AUC-PR Gain gives an improvement over to traditional PR analysis by introducing normalized gain metrics that enable more meaningful model comparisons (Flach and Kull, 2015).

Threshold-Dependent Performance Metrics. Classification metrics in binary prediction tasks inherently depend on the chosen decision threshold. This dependency becomes particularly crucial in cost-sensitive learning scenarios and imbalanced datasets, where optimal thresholds may vary significantly across different models. Our comprehensive analysis of threshold optimization techniques, detailed in Section 7.3, addresses these challenges. This methodological approach ensures equitable model comparisons while reflecting real-world operational requirements. The systematic examination of threshold optimization not only strengthens the validity of our experimental results but also contributes to the broader discourse on performance evaluation in group fairness and robustness assessment. The prominent threshold-dependent performance metrics used in our experiments are the following (i) Balanced Accuracy (ii) F1 Score (iii) Matthews Correlation Coefficient (MCC) (iv) Precision (v) Recall.

Group Fairness Metrics. We utilized different group fairness metrics to evaluate our technique. We used the fairness metrics that are implemented in FAIR FAIRNESS BENCHMARK (FFB) (Han et al., 2024). FFP specifically designed to evaluate different in-processing debiasing methods. In our experiments, we have used the following metrics: (i) Equality of Opportunity (eopp) (ii) Demographic Parity (dp) (iii) p-Rule (prule) (iv) Equalized Odds (eodd) (iv) ROC AUC Parity (aucp) (v) Balance for Negative Class (bfn) (vi) Balance for Positive Class (bfp) (vii) Area Between CDF Curves (abcc). (Han et al., 2024).

Fairness Metrics Notation. For the group fairness metrics we adopt a systematic notation that distinguishes between probability-based and threshold-based evaluations. When utilizing output probability estimates, we denote demographic parity, equal opportunity, equalized odds, and p-Rule as *dpe*, *eoppe*, *eodde*, and *prulee*, respectively. Conversely, when evaluating binary predictions derived from threshold-based classification, these metrics are denoted as *dp*, *eopp*, *eodd*, and *prule*. This notational convention aligns with the experimental framework and code implementation used in FFB (Han et al., 2024), providing consistency in metric interpretation across probability and binary domains.

Nuanced Metrics. Nuanced metrics subgroup-AUC, BPSN-AUC, and BNSP-AUC (Borkan, Dixon, Li, Sorensen, Thain and Vasserman, 2019; Borkan, Dixon, Sorensen, Thain and Vasserman, 2019) provide a threshold-agnostic assessment in machine learning models. Those metrics are used to identify various types of biases. They divide the data into two subgroups: (i) one representing groups which contains both positive and non-positive elements and (ii) another representing a background group. Specifically, the subgroup-AUC, BPSN-AUC, and BNSP-AUC were used to measure the bias minimization performance of the model for individual identity subgroups on datasets, CelebA, UTKFace.

Worst-Group Accuracy Metric. It is defined as the lowest accuracy observed across all the groups. A higher worst-group accuracy value suggests the machine learning models are less likely to mistakenly associate demographic identities with toxicity (Koh et al., 2021).

7 Experimental Setting

7.1 Neural Network Architecture

For our experimental framework, we adopt the ResNet-18 architecture (He et al., 2016) as the backbone network, following the implementation detailed in Fair Fairness Benchmark (FFB) (Han et al., 2024). This architecture serves as the foundation for all experiments conducted on the CelebA and UTKFace image datasets with ERM, GLCS and DiffEopp. For the CivilComments-WILDS dataset, our proposed GLCS framework leverages the BERT model (Devlin, 2018). The baseline comparisons, including ERM, DFR, Group-DRO, JTT, and AFR, maintain consistency with the implementations specified in (Qiu et al., 2023) for the CivilComments-WILDS experiments, ensuring a fair comparative analysis.

7.2 Early Stopping Criterion

The challenge of determining optimal stopping criteria in fair machine learning is particularly complex due to the inherent trade-offs between multiple competing objectives: group robustness, model utility, and fairness metrics. This critical aspect of training has received limited attention in the existing literature. Han et al. (2024) proposed a deterministic stopping strategy in their Fair Fairness Benchmark (FFB) framework based on learning rate decay. In contrast, Sulaiman et al. (2024) employed a more empirical approach in their work as follows: (1) monitor model performance on the validation set. (2) evaluate multiple metrics simultaneously (utility and fairness metrics). (3) Stop training when a satisfactory trade-off is achieved within early epochs. In our experiments, we follow the approach proposed by Sulaiman et al. (2024).

7.3 Classification Thresholds in the Experimental Datasets:

Binary classification in deep learning confronts significant challenges when applied to imbalanced datasets, where conventional threshold-setting strategies fail to capture nuanced distributional complexities. For example, CelebA dataset exemplifies this critical challenge, presenting a stark class distribution disparity with 32% positive instances (51,982 samples) against 68% negative instances (110,788 samples), systematically challenging traditional machine learning paradigms. Our empirical analysis reveals the inherent limitations of the standard 0.5 threshold, which presupposes uniform class representation—a premise fundamentally misaligned with real-world data characteristics. By recalibrating the classification threshold to approximately 0.32, we demonstrate a principled approach to mitigating class imbalance that enhances minority class sensitivity and improves overall predictive performance. Moreover, our Group-Level Cost-Sensitive (GLCS) approach introduces a sophisticated probabilistic framework that recalibrates class boundaries, fundamentally challenging traditional binary classification paradigms. Unlike conventional methods, our approach explicitly accommodates group-level heterogeneity by implementing constraint mechanisms that transform class separation strategies across distinct demographic or feature-based subgroups. By developing a flexible threshold optimization strategy, we enable a more granular and contextually responsive machine learning model that can adjust its decision boundaries to reflect the intricate complexities of real-world data representations. Therefore, the selection of an optimal classification threshold necessitates a sophisticated multi-dimensional analysis that integrates statistical techniques such as receiver operating characteristic (ROC) curve evaluation, F1 score maximization, and precision-recall curve assessment. Domain-specific considerations fundamentally modulate threshold selection—for instance, medical diagnostics prioritize sensitivity to minimize false negatives, while cybersecurity applications might emphasize precision to mitigate false positive risks.

7.3.1 Implementation and Empirical Methodology. Our rigorous threshold optimization framework is underpinned by the sophisticated `binclass-tools` package[1]

[1] https://github.com/lucazav/binclass-tools

on both CelebA and UTKFace datasets, a comprehensive computational toolkit designed for advanced binary classification analysis. The implementation follows a meticulously structured empirical protocol that systematically addresses the complexities of threshold optimization across diverse machine-learning models. We commence by training models using multiple approaches, including established baselines and our proposed GLCS, which enables a comprehensive comparative analysis. The methodology involves generating nuanced probability distributions for each model, allowing for granular insight into predictive performance characteristics. Leveraging the `binclass-tools` package, we use it to determine optimal classification thresholds. Our evaluation protocol rigorously assesses model performance using these optimized thresholds, employing consistent and theoretically grounded evaluation criteria to ensure methodological integrity. This systematic approach not only facilitates a fair and comprehensive comparison across different methodological approaches but also maintains a principled framework for cost-sensitive learning and group fairness.

7.4 Calibrating Neural Networks.

Deep neural networks have demonstrated remarkable discriminative performance across various tasks; however, their probability estimates often lack proper calibration, potentially leading to overconfident or underconfident predictions. A well-calibrated model should produce probability estimates that reflect true empirical frequencies—for instance, among predictions with confidence of 0.8, approximately 80% should be correctly classified. Calibration is particularly crucial in high-stakes applications where reliable uncertainty quantification is essential for decision-making processes.

In our experiments, we employ Temperature Scaling (Guo et al., 2017), a simple yet effective post-processing calibration technique that can be applied to the logits while preserving the model's discriminative capabilities. Given our model's output probabilities $p_i \in [0, 1]$, we first convert these to logits through the inverse sigmoid function: $z_i = \log(\frac{p_i}{1-p_i})$. Temperature scaling then modifies these logits by introducing a temperature parameter $T > 0$, and the calibrated probabilities are computed as follows:

$$\hat{p}_i = \sigma(z_i/T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \quad T > 0$$

The optimal temperature parameter T^* is learned by minimizing the negative log-likelihood (NLL) on a held-out validation set. This approach is particularly advantageous as it maintains the model's ranking performance metrics (e.g., AUC-ROC) due to the monotonic nature of the temperature scaling operation. Furthermore, the optimization of a single parameter T reduces the risk of overfitting compared to more complex calibration methods, while effectively addressing both under-confidence and over-confidence in model predictions. Our experimental results demonstrate that this calibration method successfully improves the reliability of probability estimates while maintaining the model's discriminative performance and fairness properties. This is particularly important as well-calibrated probabilities enable more reliable decision-making processes and better interpretation of model confidence.

7.5 GLCS Methodology with CivilComments-WILDS Dataset

Our comprehensive research methodology systematically addresses group fairness challenges in the CivilComments-WILDS dataset through a carefully designed experimental protocol. We commenced by conducting an initial baseline evaluation using Empirical Risk Minimization (ERM) to train our model, which enabled us to comprehensively assess the baseline performance and identify group-specific disparities, critically revealing the Christian demographic group as experiencing the most significant accuracy degradation (Section 8.3.2). Leveraging these insights, we subsequently applied our proposed approach (GLCS) with a targeted intervention focused on improving fairness specifically for the underperforming Christian group. To validate the effectiveness of our approach, we performed a rigorous comparative analysis, benchmarking our GLCS method against state-of-the-art baseline techniques commonly employed for enhancing group robustness (Section 5.5). This methodical approach allows us to systematically demonstrate the potential of our proposed method in mitigating group-based performance disparities within complex machine learning fairness challenges and improving group robustness.

7.6 Hyperparameters

In this section, we elucidate the hyperparameter configurations employed across different datasets, highlighting our methodical approach to parameter selection.

CelebA and UTKFace Datasets. For the GLCS framework proposed in Section 4, we selected hyperparameters based on the unique characteristics of each dataset. On the CelebA dataset, characterized by substantial intersectional distribution disparities (as evidenced in Table 1), we employed $\mu_1 = \mu_2 = \mu_3 = 0.5$ and $\delta = 0.2$. Conversely, the UTKFace dataset, exhibiting minimal group distribution variations (detailed in Table 2), warranted a more nuanced approach with $\mu_1 = \mu_2 = \mu_3 = \epsilon$ and $\delta = 0.1$, where $\epsilon \approx 0$. For comparative methods, namely ERM and DiffEopp, we consistently utilized the hyperparameters established in the Fair Fairness Benchmark (FFB) (Han et al., 2024) across both datasets.

CivilComments-WILDS Dataset. In the context of the CivilComments-WILDS dataset, we configured the GLCS method with $\mu_1 = \mu_2 = \mu_3 = 1$ and $\delta = 0.2$. For comparative methods, we utilized the hyperparameters for Just Train Twice (JTT) as specified in (Liu et al., 2021) and those for ERM, AFR, DFR, and Group-DRO following (Qiu et al., 2023). Consistent with prior work (Qiu et al., 2023; Liu et al., 2021), we applied the standard 0.5 threshold for metric evaluation across all methods.

8 Results and Analysis

We present a comprehensive analysis of our proposed GLCS approach compared with baseline methods on three datasets: CelebA, UTKFace, and CivilComments-WILDS. This comparison enables us to assess the efficiency of GLCS and Calibrated GLCS (after applying Temperature Scaling to our GLCS model) in achieving a balanced trade-off

between model performance and fairness objectives. The following subsections explain our findings for each dataset.

8.1 Experimental Evaluation on CelebA Dataset

8.1.1 Analysis of Threshold-Agnostic Performance Metrics. Our experimental evaluation on the CelebA dataset reveals notable performance variations across the different methodologies (Table 4). The ERM approach demonstrates superior performance across all metrics, achieving the highest ROC AUC (0.8576), AUC-PR Gain (0.7911) and PR AUC (0.7913), while maintaining the lowest Brier Score (0.1475). The DiffEopp method shows a considerable performance decline, with ROC AUC, AUC-PR Gain and PR AUC dropping to 0.7815, 0.6008 and 0.6011, respectively, though maintaining a relatively competitive Brier Score of 0.1861. Both GLCS and Calibrated GLCS exhibit identical discriminative capabilities, with ROC AUC of 0.8443, AUC-PR Gain of 0.7369 and PR AUC of 0.7371, positioning them as intermediate solutions between ERM and DiffEopp. However, they differ significantly in their calibration performance, with Calibrated GLCS achieving a substantially better Brier Score (0.2044) compared to standard GLCS (0.3349). The Calibrated GLCS emerges as a particularly promising approach, demonstrating robust discriminative capabilities while significantly enhancing probability calibration compared to its uncalibrated variant. Furthermore, both Calibrated GLCS and GLCS exhibit comparable fairness characteristics across various fairness metrics (discussed in Section 8.1.5). This comprehensive evaluation suggests that these methods achieve an optimal balance between maintaining competitive predictive performance and satisfying fairness constraints, positioning them as viable solutions for applications where both accuracy and fairness are crucial considerations. The empirical evidence particularly favors the Calibrated GLCS variant. It preserves the fairness properties of the base GLCS while providing more reliable probability estimates as shown by its improved Brier Score and AUC-PR Gain.

Metric	ERM	DiffEopp	GLCS	Calibrated GLCS
ROC AUC ↑	0.8576	0.7815	0.8443	0.8443
PR AUC ↑	0.7913	0.6011	0.7371	0.7371
Brier Score ↓	0.1475	0.1861	0.3349	0.2044
AUC-PR Gain ↑	0.7911	0.6008	0.7369	0.7369

Table 4. Invariant Performance Metrics for different methods on Celeb-A Dataset

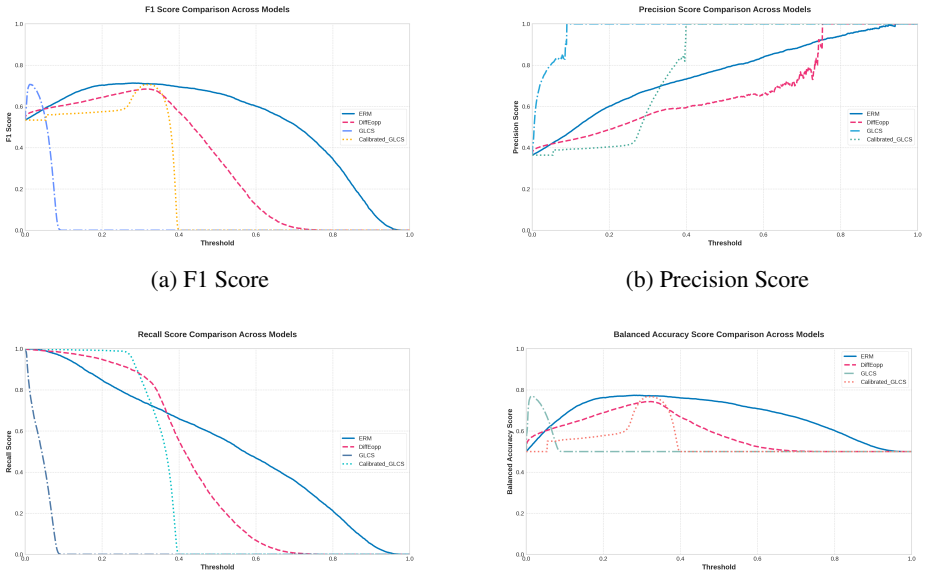
8.1.2 Analysis of Threshold-Dependent Performance Metrics. Our experimental results in Table 5 demonstrate notable performance variations across different methodological approaches. The baseline ERM achieves the highest F1 score (0.7132) at a threshold of 0.281. Moreover, ERM achieves balanced accuracy (0.7737), establishing a strong performance benchmark. The Calibrated GLCS shows comparable performance

metrics (balanced accuracy: 0.7715, F1 score: 0.7100), at a threshold of 0.315, while incorporating the proposed constraints. This minimal performance trade-off is particularly noteworthy, as using constraints typically incurs more substantial accuracy penalties, as we see for DiffEopp method. The DiffEopp method exhibits the highest recall (0.8601) but the lowest precision (0.5680), indicating a potential bias toward positive predictions. The GLCS method's very low threshold (0.012) for best F1 score (0.7672) compared to other methods (ranging from 0.281 to 0.315) confirms our hypothesis about probability space compression. This is effectively addressed through calibration as evidenced by the Calibrated GLCS's threshold restoration to 0.315. The Matthews Correlation Coefficient, a particularly robust metric for imbalanced datasets, shows consistent ranking with F1 scores, with ERM and Calibrated GLCS achieving the highest values (0.5341 and 0.5357, respectively).

Method	**Threshold**	**Balanced Accuracy**	**F1 Score**	**Matthews Corr. Coef.**	**Precision**	**Recall**
ERM	0.281	0.7737	0.7132	0.5341	0.6678	0.7652
DiffEopp	0.313	0.7428	0.6842	0.4698	0.5680	0.8601
GLCS	0.012	0.7672	0.7066	0.5161	0.6345	0.7972
Calibrated GLCS	0.315	0.7715	0.7100	0.5357	0.6864	0.7354

Threshold selected for optimal F1 score across different methods.
Table 5. Performance Metrics Comparison on Celeb-A Dataset

8.1.3 Performance Metrics Across Threshold Spectrum. Our comprehensive investigation of threshold sensitivity reveals nuanced performance characteristics across different fairness-aware machine learning models. The ERM model demonstrates significant robustness, maintaining F1 scores above 0.6 across a broad threshold range (0.2–0.8), in stark contrast to DiffEopp and Calibrated GLCS, which exhibit sharp performance degradation beyond their optimal threshold regions (Figure 1a). The base GLCS model displays an exceptionally narrow operational range, suggesting significant compression induced by its fairness constraints. Balanced accuracy analysis (Figure 1d) reveals a consistent peak within the 0.2–0.4 threshold range for all models, except GLCS, with ERM showcasing the most gradual performance decline at higher thresholds. The recall-precision trade-off curves (Figures 1b, 1c) illuminate the models' distinct behavioral patterns: ERM maintains the most balanced transition, while GLCS models exhibit more abrupt shifts, particularly in recall sensitivity. The wide threshold variations, especially in the GLCS approach, provide critical insights into the mechanisms of fairness-aware model design in GLCS framework. The probability space compression appears to stem from simultaneously satisfying multiple group-level fairness constraints, with the interaction between fairness penalties and base loss fun-



(a) F1 Score (b) Precision Score

(c) Recall Score (d) Balanced Accuracy Score

Fig. 1. Performance Metrics Across Threshold Spectrum on CelebA Dataset

damentally reshaping the model's decision boundaries. Our investigation unveils that hyperparameters μ_1, μ_2, μ_3, and δ substantially govern this probability space compression, with incremental parametric adjustments potentially yielding significant shifts in distributional representation based on different thresholds, as we meticulously illustrate in our UTKFace dataset analysis (Section 8.2).

The Calibrated GLCS's restoration of a more standard threshold through temperature scaling demonstrates an elegant solution—effectively "decompressing" the probability space while preserving the model's discriminative and fairness characteristics. This calibration approach not only broadens the model's robust performance region but also provides a promising strategy for maintaining fairness without sacrificing predictive consistency across different threshold values.

8.1.4 Subgroup Performance Analysis using Nuanced Metrics. Our empirical evaluation in Table 6 reveals significant variations in performance across methods and gender subgroups in the CelebA dataset. The baseline ERM demonstrates strong discriminative power with Subgroup-AUC scores of 0.805 and 0.831 for male and female subgroups, respectively, indicating robust within-group classification capabilities. However, the stark contrast between BPSN-AUC (0.958) and BNSP-AUC (0.512) metrics indicates substantial asymmetry in cross-group performance, suggesting potential systematic biases in the model's decision boundary. The DiffEopp method achieves more balanced cross-group metrics (BPSN-AUC: 0.873, BNSP-AUC: 0.615). This enhanced fairness, however, comes at the cost of reduced within-group performance, particularly

for the female subgroup, where Subgroup-AUC drops to 0.714. This trade-off exemplifies the challenging balance between fairness and performance objectives. Notably, our GLCS, including its calibrated variant, maintains strong within-group performance (male: 0.795, female: 0.807) while exhibiting cross-group behavior similar to ERM (BPSN-AUC: 0.956, BNSP-AUC: 0.487). The preservation of high female Subgroup-AUC (0.807) is particularly noteworthy, as it represents only a 2.9% decrease from ERM while incorporating the proposed constraints for fairness in GLCS framework. Our results validate that GLCS effectively maintains discriminative power while working within the fairness framework. The substantial disparity in subgroup sizes (male: 7,715, female: 12,247) adds another dimension to these findings, highlighting the importance of considering demographic imbalance in fairness-aware model development. The analysis underscores the complex interplay between maintaining strong predictive performance and achieving equitable treatment across demographic subgroups.

Method	**Subgroup**	**Subgroup Size**	**Subgroup AUC**	**BPSN AUC**	**BNSP AUC**
ERM	Male	7,715	0.8050	0.9583	0.5118
	Female	12,247	0.8307	0.5118	0.9583
DiffEopp	Male	7,715	0.7976	0.8734	0.6150
	Female	12,247	0.7142	0.6150	0.8734
GLCS & Calibrated GLCS	Male	7,715	0.7954	0.9564	0.4866
	Female	12,247	0.8067	0.4867	0.9564

Table 6. Nuanced Metrics on CelebA Dataset

8.1.5 Empirical Analysis of Fairness Metrics using CelebA Dataset. Our empirical evaluation of algorithmic fairness methodologies reveals critical insights into the performance of ERM, DiffEopp, GLCS, and Calibrated GLCS across multiple fairness dimensions (Table 7). The GLCS approach emerges as a standout winner, consistently demonstrating superior fairness metrics across various evaluation criteria. Notably, GLCS achieves exceptional results in minimizing opportunity disparities (eoppe), with an error rate of 2.84%, significantly outperforming both DiffEopp (4.60%) and the baseline ERM (30.78%). The equalized odds analysis (eodde) further substantiates GLCS's effectiveness, revealing minimal error (4.20%) compared to Calibrated GLCS (12.80%), DiffEopp (18.07%), and ERM (47.25%), which underscores its robust capability in maintaining fairness across positive and negative outcome scenarios.

The Demographic Parity and distribution divergence metrics provide additional validation of GLCS's approach. With a Demographic Parity (dpe) of 2.51%, GLCS significantly surpasses DiffEopp (15.97%) and ERM (29.38%), demonstrating its abil-

ity to ensure equitable prediction distributions across demographic groups. The Calibrated GLCS variant further enhances these results, achieving a p-rule score (prulee) of 72.74% and maintaining minimal Area Between CDF Curves (9.05%), compared to DiffEopp's 15.97% and ERM's 29.38%. The ROC AUC Parity (aucp) analysis reveals remarkably consistent performance, with GLCS and Calibrated GLCS showing minimal disparities (1.13%), in stark contrast to ERM's 2.57% and DiffEopp's 8.34% variations.

The examination of balanced for positive (bfp) class and negative class (bfn) reveals a performance hierarchy. GLCS achieves the most optimal error balance with bfp at 2.84% and bfn at 1.36%, demonstrating minimal classification disparities. Calibrated GLCS maintains strong performance with 5.24% bfp and 7.56% bfn, while DiffEopp shows moderate imbalance (4.60% bfp, 13.47% bfn). The baseline ERM approach exhibits the most significant error disparities, with 30.78% bfp and 16.47% bfn, highlighting the critical importance of fairness-aware methodological interventions.

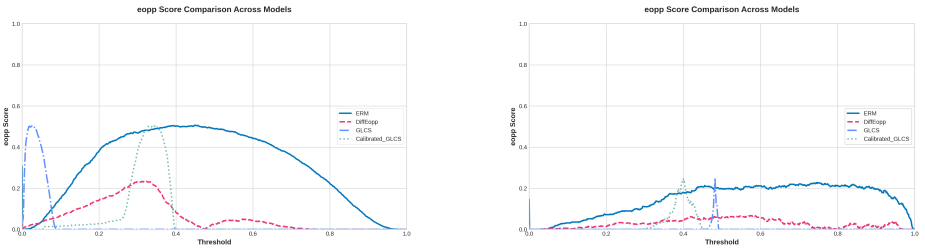
Metric	ERM	DiffEopp	GLCS	Calibrated GLCS
p-Rule (prulee) ↑	31.67	56.76	22.68	72.74
Equal Opportunity (eoppe) ↓	30.78	4.60	2.84	5.24
Equalized Odds (eodde) ↓	47.25	18.07	4.20	12.80
Demographic Parity (dpe) ↓	29.38	15.97	2.51	9.05
Balance for Positive Class (bfp) ↓	30.78	4.60	2.84	5.24
Balance for Negative Class (bfn) ↓	16.47	13.47	1.36	7.56
ROC AUC Parity (aucp) ↓	2.57	8.34	1.13	1.13
Area Between CDF Curves (abcc) ↓	29.38	15.97	2.51	9.05

Table 7. Calculate various fairness metrics with CelebA Datase

Threshold Sensitivity Analysis of Equal Opportunity. Figure 2a illustrates the Equal Opportunity (eopp) score variations across different classification thresholds for ERM, DiffEopp, GLCS, and Calibrated GLCS models. The analysis reveals several notable patterns: ERM exhibits the highest sensitivity to threshold selection, with eopp scores peaking at approximately 0.5 around the 0.4 threshold and gradually declining toward both extremes. In contrast, both GLCS and Calibrated GLCS demonstrate remarkable stability across most threshold values, maintaining consistently low eopp scores (< 0.1) except for a brief spike in GLCS at very low thresholds (< 0.1). DiffEopp shows intermediate performance with moderate threshold sensitivity, reaching a maximum eopp score of approximately 0.25 around 0.35 threshold. Notably, Calibrated GLCS exhibits a localized increase in eopp score around the 0.35 threshold region but quickly returns to stable performance. This comprehensive analysis suggests that GLCS and Calibrated GLCS provide more robust and threshold-invariant fairness guarantees compared to

traditional ERM and DiffEopp approaches, making them more reliable choices for applications requiring consistent fairness across different operating points.

8.1.6 Performance and Equal Opportunity Trade-Off on CelebA Dataset. Our comprehensive experimental evaluation unveils intricate trade-offs between predictive performance and fairness metrics across heterogeneous threshold configurations on CelebA Dataset. To ensure a rigorous and fair comparative analysis, we employ threshold-agnostic metrics: AUC-PR Gain and the threshold-agnostic Equal Opportunity Difference (eoppe) metric. The proposed GLCS and Calibrated GLCS approaches demonstrate remarkable fairness characteristics, consistently exhibiting substantially lower Equal Opportunity Difference scores. Specifically, the Calibrated GLCS achieved an eoppe of 5.24, while the GLCS method realized an eoppe of 2.84, in stark contrast to the Empirical Risk Minimization (ERM) baseline (eoppe = 30.78) and the DiffEopp approach (eoppe = 4.60). Notably, these improved fairness metrics are attained without compromising predictive performance. Both GLCS variants maintained competitive AUC-PR Gain scores (0.7369), comparable to DiffEopp (0.6008) and ERM (0.7911). This empirical evidence suggests that the proposed GLCS methodologies offer a principled approach to mitigating discriminatory outcomes while preserving high-fidelity predictive precision.



(a) eopp Score on CelebA (b) eopp Score on UTKFace

Fig. 2. eopp Metric Across Threshold Spectrum on CelebA and UTKFace

8.2 Experimental Evaluation on UTKFace Dataset

8.2.1 Analysis of Threshold-Agnostic Performance Metrics Our experimental evaluation on the UTKFace dataset reveals interesting performance patterns across the different methodologies (Table 8). The ERM approach maintains its superior performance across all metrics, achieving the highest ROC AUC (0.9015), AUC-PR Gain (0.8983) and PR AUC (0.8993), while demonstrating the lowest Brier Score (0.1266). Although DiffEopp shows slightly reduced performance compared to ERM, it still maintains strong discriminative capabilities with ROC AUC of 0.8754, AUC-PR Gain of 0.8599 and PR AUC of 0.8609, along with a competitive Brier Score of 0.1459. Both GLCS

and Calibrated GLCS exhibit identical discriminative performance, with ROC AUC of 0.8866, AUC-PR Gain of 0.8741 and PR AUC of 0.8752, positioning them between ERM and DiffEopp in terms of predictive capability. However, they differ in their calibration performance, with Calibrated GLCS achieving a better Brier Score (0.2276) compared to standard GLCS (0.2410). The Calibrated GLCS emerges as a particularly promising approach, demonstrating robust discriminative capabilities while enhancing probability calibration compared to its uncalibrated variant.

Metric	ERM	DiffEopp	GLCS	Calibrated GLCS
ROC AUC ↑	0.9015	0.8754	0.8866	0.8866
PR AUC ↑	0.8993	0.8609	0.8752	0.8752
Brier Score ↓	0.1266	0.1459	0.2410	0.2276
AUC-PR Gain ↑	0.8983	0.8599	0.8741	0.8741

Table 8. Invariant Performance Metrics for different methods on UTKFace Dataset

8.2.2 Analysis of Threshold-Dependent Performance Metrics. Our experimental results in Table 9 demonstrate notable performance variations across different methodological approaches based on the threshold for best F1 score for each approach. The baseline ERM achieves superior performance across multiple metrics, including the highest balanced accuracy (0.8069), F1 score (0.8010), and precision (0.7606) at a threshold of 0.406. The Calibrated GLCS demonstrates remarkably competitive performance (balanced accuracy: 0.8039, F1 score: 0.7997) at a threshold of 0.385. This minimal performance trade-off is particularly noteworthy, as some constraints typically incur more substantial accuracy penalties, as evidenced by the DiffEopp method's performance. While DiffEopp exhibits the highest recall (0.8614), it shows the lowest performance across other metrics, including precision (0.7248) and F1 score (0.7873), suggesting a potential bias toward positive predictions. The GLCS method achieves intermediate performance levels (balanced accuracy: 0.7950, F1 score: 0.7925) with a notably higher threshold (0.479) compared to other methods. Importantly, the Matthews Correlation Coefficient, which is particularly robust for imbalanced datasets, confirms the relative performance ordering, with ERM and Calibrated GLCS achieving the highest values (0.6128 and 0.6075, respectively). These results suggest that Calibrated GLCS effectively maintains strong predictive performance while incorporating the proposed constraints, making it a promising approach for applications requiring both accuracy and fairness considerations.

8.2.3 Performance Metrics Across Threshold Spectrum. Our examination of threshold sensitivity across models (Figure 3) reveals several distinctive behavioral patterns and performance characteristics. The F1 score analysis (Figure 3a) demonstrates that ERM and DiffEopp models maintain robust performance across a broad threshold range

Metric	ERM	DiffEopp	GLCS	Calibrated GLCS
Balanced Accuracy ↑	0.8069	0.7881	0.7950	0.8039
F1 Score ↑	0.8010	0.7873	0.7925	0.7997
Matthews CC ↑	0.6128	0.5782	0.5908	0.6075
Precision ↑	0.7606	0.7248	0.7364	0.7510
Recall ↑	0.8460	0.8614	0.8578	0.8551
Best Threshold	0.406	0.452	0.479	0.385

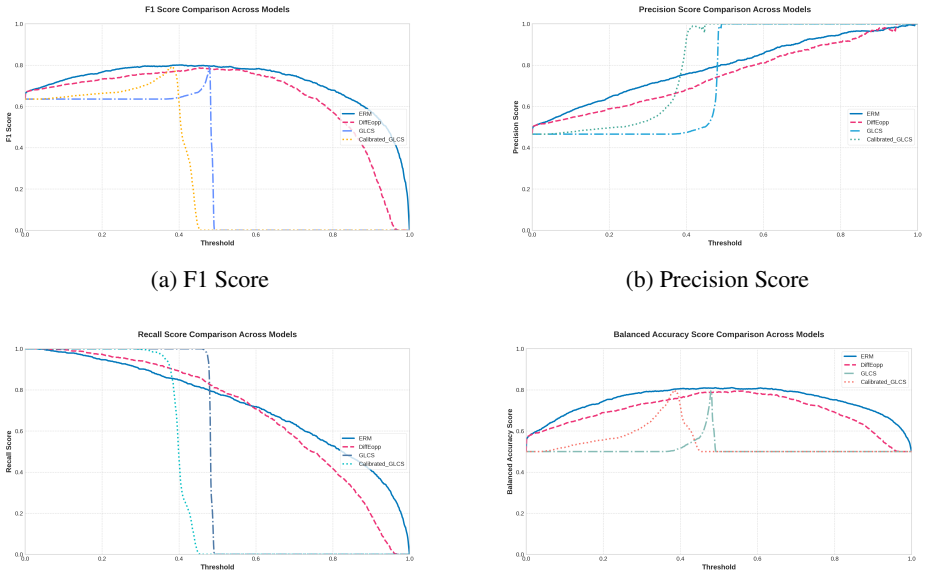
Table 9. Threshold for best F1 score for different methods on UTKFace Dataset with the corresponding variant performance metrics

(0.2-0.6), consistently achieving F1 scores above 0.6. While both GLCS and Calibrated GLCS exhibit comparable peak performance, they show more pronounced degradation outside their optimal threshold regions (approximately 0.37-0.5). The base GLCS displays a distinctive sharp performance spike near the 0.48 threshold, indicating a highly concentrated probability distribution. In terms of balanced accuracy trends (Figure 3d), the ERM baseline achieves the highest overall performance, reaching and maintaining a balanced accuracy of approximately 0.8 across the threshold range of 0.4-0.6, with graceful degradation at extreme thresholds. DiffEopp demonstrates comparable but slightly lower performance, maintaining balanced accuracy scores around 0.75-0.78 in the optimal range (0.4-0.6), though showing more pronounced degradation at higher thresholds compared to ERM. The GLCS and Calibrated GLCS methods exhibit notably different behaviors: Calibrated GLCS shows a gradual improvement up to threshold 0.4, reaching a peak of approximately 0.78, followed by an abrupt performance drop, while GLCS maintains a constant lower performance (around 0.5) before displaying a sharp, localized spike to 0.8 at threshold around 0.48.

This comparative analysis suggests that while fairness-oriented approaches like DiffEopp can achieve near-baseline performance, they may introduce some performance trade-offs, particularly in threshold sensitivity. The distinct behavioral patterns of GLCS variants indicate potential stability challenges in their balanced accuracy maintenance across different classification thresholds.

The recall-precision trade-off analysis (Figures 3b, 3c) reveals that recall curves demonstrate the expected monotonic decrease with increasing threshold, accompanied by corresponding increases in precision. ERM maintains the most gradual transition between these metrics, indicating superior calibration. Both GLCS variants exhibit step-like transitions around their respective optimal thresholds (0.48 for base GLCS, 0.39 for Calibrated GLCS), suggesting binary-like behavior in their predictions. Notably, DiffEopp maintains higher recall but lower precision compared to other models across most thresholds.

8.2.4 Subgroup Performance Analysis using Nuanced Metrics. Our empirical evaluation in Table 10 reveals noteworthy patterns in performance across methods and gender subgroups in the UTKFace dataset. The baseline ERM demonstrates strong discriminative power with Subgroup-AUC scores of 0.8927 and 0.9015 for male and female



(a) F1 Score (b) Precision Score

(c) Recall Score (d) Balanced Accuracy Score

Fig. 3. Performance Metrics Across Threshold Spectrum on UTKFace Dataset

subgroups respectively, indicating robust within-group classification capabilities. The complementary relationship between BPSN-AUC (0.9578) and BNSP-AUC (0.7946) metrics suggests a moderate asymmetry in cross-group performance.

The DiffEopp method achieves 0.8950 for BPSN-AUC (male subgroup) and 0.8446 for BNSP-AUC (female subgroup). This comes with a modest trade-off in within-group performance, with Subgroup-AUC slightly decreasing to 0.8723 and 0.8717 for male and female subgroups, respectively.

Our proposed GLCS approach and its calibrated variant maintain strong Subgroup AUC performance (male: 0.8874, female: 0.8760) while showing cross-group behavior similar to ERM (BPSN-AUC: 0.9521, BNSP-AUC: 0.7654). The relatively balanced subgroup sizes (male: 1131, female: 1239) in the UTKFace dataset provide a more equitable basis for evaluation compared to more imbalanced datasets.

These results demonstrate that while GLCS effectively maintains discriminative power within the fairness framework, the challenge of cross-group prediction asymmetry persists, albeit to a lesser degree than in comparable datasets. This analysis highlights the delicate balance between maintaining strong predictive performance and achieving equitable treatment across demographic subgroups, even in relatively balanced dataset conditions.

8.2.5 Empirical Analysis of Fairness Metrics using UTKFace Dataset. Our comprehensive empirical evaluation of algorithmic fairness on the UTKFace dataset presents a comparative analysis of the four methodologies: ERM, DiffEopp, GLCS, and Cal-

Method	Subgroup	Subgroup AUC	BPSN AUC	BNSP AUC	Size
ERM	Male	0.8927	0.9578	0.7946	1131
	Female	0.9015	0.7946	0.9578	1239
DiffEopp	Male	0.8723	0.8950	0.8446	1131
	Female	0.8717	0.8446	0.8950	1239
GLCS & Calibrated GLCS	Male	0.8874	0.9521	0.7654	1131
	Female	0.8760	0.7654	0.9521	1239

Table 10. Nuanced Metrics for different methods on UTKFace Dataset

ibrated GLCS. The results, presented in Table 11, reveal significant findings across multiple fairness dimensions. The evaluation demonstrates that GLCS achieves exceptional performance in minimizing *Equal Opportunity* disparities (eoppe), with an error rate of 0.23%, significantly outperforming DiffEopp (2.16%) despite the latter being specifically designed for this criterion (Equal Opportunity). Calibrated GLCS maintains strong performance with 1.25% for eoppe, while the baseline ERM exhibits substantially higher disparity (14.61%).

Analysis of the *Equalized Odds* reveals a clear performance hierarchy, with GLCS achieving optimal fairness at 0.65% for eodde, followed by Calibrated GLCS demonstrating strong fairness capability at 3.28%. DiffEopp shows improved performance with 7.09% for eodde, while ERM exhibits the highest disparity at 27.24%. These findings underscore GLCS's superior capability in maintaining fairness across outcome scenarios. Furthermore, GLCS demonstrates superior performance with a *p-rule* score of 98.69%, significantly outperforming both Calibrated GLCS (92.01%), DiffEopp (80.22%) and ERM (61.78%). This metric indicates that GLCS effectively maintains balanced probability distributions for positive and negative outcomes across demographic groups.

The analysis of *Demographic Parity* (dpe) reveals substantial variations, with GLCS achieving remarkable performance with a dpe of 0.62%, significantly surpassing Calibrated GLCS (3.00%), DiffEopp (10.69%) and ERM (22.03%). This demonstrates GLCS's effectiveness in ensuring equitable prediction distributions across demographic groups.

In terms of error rate balance, analysis of *Balance for Positive Class* (bfp) and *Balance for Negative Class* (bfn) shows that GLCS achieves optimal balance (bfp: 0.23%, bfn: 0.42%), while Calibrated GLCS maintains strong balance (bfp: 1.25%, bfn: 2.03%). DiffEopp exhibits moderate imbalance (bfp: 2.16%, bfn: 4.93%), and ERM shows significant disparity (bfp: 14.61%, bfn: 12.63%). Moreover, in terms of ROC AUC Parity (aucp), DiffEopp demonstrates minimal disparities (0.06%) in predictive performance across demographic groups, outperforming ERM (0.88%). Both GLCS and Calibrated GLCS show slightly higher but consistent disparities (1.15%). The Area Between CDF Curves (abcc) metric further validates GLCS's effectiveness, achieving minimal distribution divergence (0.62%), followed by Calibrated GLCS (3.00%). DiffEopp (10.72%) and ERM (22.03%) exhibit substantially higher disparities.

Metric	ERM	DiffEopp	GLCS	Calibrated GLCS
p-Rule (prulee) ↑	61.78	80.22	98.69	92.01
Equal Opportunity (eoppe) ↓	14.61	2.16	0.23	1.25
Equalized Odds (eodde) ↓	27.24	7.09	0.65	3.28
Demographic Parity (dpe) ↓	22.03	10.69	0.62	3.00
Balance for Positive Class (bfp) ↓	14.61	2.16	0.23	1.25
Balance for Negative Class (bfn) ↓	12.63	4.93	0.42	2.03
ROC AUC Parity (aucp) ↓	0.88	0.06	1.15	1.15
Area Between CDF Curves (abcc) ↓	22.03	10.72	0.62	3.00

Table 11. Calculate various fairness metrics with UTKFace Dataset

Threshold Sensitivity Analysis of Equal Opportunity. Figure 2b demonstrates the superior fairness characteristics of GLCS-based approaches compared to alternative methods. While the ERM baseline exhibits persistent unfairness with eopp scores steadily increasing to approximately 0.2 across the 0.4-0.8 threshold range, and DiffEopp showing moderate improvement with scores around 0.05, both GLCS and Calibrated GLCS demonstrate remarkable fairness preservation across most threshold values, maintaining near-zero eopp scores throughout the majority of the threshold spectrum. The tiny elevation in eopp scores around threshold 0.4 for these methods can be interpreted as a controlled trade-off point where the models actively adjust their decision boundaries to maintain long-term fairness stability. This localized behavior suggests a sophisticated fairness optimization strategy, where the models temporarily accept a minor fairness deviation to establish robust equilibrium across the broader threshold range. Particularly noteworthy is how both GLCS variants achieve nearly perfect Equal Opportunity (eopp $\approx$ 0) across extensive threshold regions (0.0-0.35 and 0.45-1.0), demonstrating their ability to maintain consistent fairness guarantees without the continuous fairness drift observed in ERM and DiffEopp. This comprehensive analysis suggests that GLCS-based approaches offer superior fairness preservation through their unique ability to establish and maintain stable equal opportunity metrics across diverse operating conditions.

8.2.6 Performance and Equal Opportunity Trade-Off on UTKFace Dataset. Our experimental evaluation reveals nuanced trade-offs between predictive performance and fairness metrics across varying threshold configurations. To ensure a rigorous and fair comparative analysis, we use threshold-agnostic metrics: AUC-PR Gain and eoppe metric. The proposed GLCS and Calibrated GLCS demonstrate remarkable fairness characteristics, consistently exhibiting substantially lower equal opportunity difference scores (eoppe). Specifically, the Calibrated GLCS achieved an eoppe of 1.25, while the GLCS method realized an eoppe of 0.23, in stark contrast to ERM baseline (eoppe = 14.61) and the DiffEopp approach (eoppe = 2.16). Notably, these improved fairness metrics are attained without compromising predictive performance for our approach.

Both GLCS variants maintained competitive AUC-PR Gain scores (0.8741), comparable to DiffEopp (0.8599) and ERM (0.8983). This empirical evidence suggests that the proposed GLCS methodologies offer a principled approach to mitigating discriminatory outcomes while preserving high-fidelity predictive precision. The results underscore the potential of GLCS methods in domains requiring stringent algorithmic fairness, particularly in high-stakes decision-making contexts where balancing performance and equitable outcomes is paramount.

8.3 Experimental Evaluation on CivilComments-WILDS Dataset

8.3.1 Analysis of Performance and Fairness Metrics. We evaluate our approach on the CivilComments-WILDS dataset using two key performance metrics: ROC AUC and Average Precision (AP), as shown in Table 12. The results demonstrate that both GLCS and ERM achieve comparable performance, with GLCS obtaining an ap of 74.62% and ROC AUC of 94.53%, while ERM achieves an ap of 74.74% and ROC AUC of 94.55%. Further analysis of fairness metrics between GLCS and ERM is presented in Table 13. The evaluation reveals that GLCS demonstrates superior performance across multiple fairness dimensions. Specifically, GLCS achieves exceptional performance in minimizing *Equal Opportunity* disparities, with (eoppe) of 4.91%, substantially outperforming ERM's 8.14%. The analysis of *Equalized Odds* reveals a clear advantage for GLCS, achieving an error rate (eodde) of 5.83% compared to ERM's 8.51%. In terms of *Demographic Parity*, GLCS demonstrates remarkable fairness with a demographic parity error (dpe) of 0.86%, significantly lower than ERM's 2.43%. Furthermore, GLCS achieves superior group fairness with a *p-rule* score of 95.85%, substantially outperforming ERM's 79.56%, indicating more balanced treatment across different demographic groups.

Metric	GLCS	ERM
Average Precision (ap)	74.62	74.74
ROC AUC	94.53	94.55

Table 12. Performance Metrics Comparison on CivilComments-WILDS Dataset

8.3.2 Analysis of Group Robustness. Our experimental results are shown in Tables 14 and 15, the baseline ERM method achieves an Average Accuracy of 92.4% but shows suboptimal performance with a Worst-Group Accuracy of 58.3% (Christian demographic group), indicating significant performance disparities across groups. By incorporating the Christian group as the sensitive feature in our proposed GLCS framework, we observe more balanced performance metrics. Specifically, GLCS achieves an Average Accuracy of 91.3% while substantially improving the Worst-Group Accuracy to 70.7%, representing a remarkable improvement of 12.4 percentage points in

Metric	GLCS	ERM
Demographic Parity Error (dpe) ↓	0.86	2.43
Equality of Opportunity Error (eoppe) ↓	4.91	8.14
Equalized Odds Error (eodde) ↓	5.83	8.51
p-Rule Error (prulee) ↑	95.85	79.56

Table 13. Fairness Metrics Comparison on CivilComments-WILDS Dataset

worst-group performance compared to ERM, with only a modest decrease of 1.1 percentage points in average accuracy. Furthermore, our experimental results demonstrate that GLCS consistently outperforms other robust baselines (DFR, Group-DRO, JTT, and AFR) on the CivilComments-WILDS dataset, establishing a new state-of-the-art in balancing average performance and group robustness on the CivilComments-WILDS dataset for this challenging benchmark.

Group	**#Samples**		**GLCS Method**		**ERM Method**	
	Non-Toxic	**Toxic**	**Non-Toxic**	**Toxic**	**Non-Toxic**	**Toxic**
Male	12,092	2,203	0.898	0.737	0.937	0.647
Female	14,179	2,270	0.912	0.725	0.946	0.640
LGBTQ	3,210	1,216	0.784	0.745	0.880	0.620
Christian	12,101	1,260	0.935	0.707	0.962	0.583
Muslim	5,355	1,627	0.820	0.744	0.903	0.607
Other religions	2,980	520	0.882	0.746	0.935	0.623
Black	3,335	1,537	0.737	0.798	0.856	0.680
White	5,723	2,246	0.760	0.784	0.866	0.660

Table 14. Group Accuracy Comparison and Sample Distribution across GLCS and ERM Methods

Method	Average Accuracy ↑	Worst-Group Accuracy ↑
ERM	0.924	0.583
DFR	0.872	0.701
Group-DRO	0.889	0.699
JTT	0.911	0.693
AFR	0.898	0.687
GLCS (Ours)	0.913	0.707

Table 15. Comparative Performanc of Group Fairness Method

9 Conclusion

This paper introduces a novel method of Group-Level Cost-Sensitive Learning (GLCS) framework, a pioneering approach that addresses critical challenges at the intersection of cost-sensitive learning, group fairness and group robustness in machine learning. By systematically incorporating group-level misclassification costs, we validate our proposed mehtod for mitigating bias while maintaining high model accuracy.

The key contributions of our work extend beyond traditional fairness interventions. We have empirically validated a fundamental synergy between group robustness and group fairness, revealing that targeted optimization strategies can simultaneously enhance model performance across underrepresented subgroups. Our approach fundamentally differs from conventional techniques by modifying the learning objective rather than artificially manipulating dataset distributions, thereby providing a more principled framework for addressing inherent biases in machine learning systems.

Our experimental results across multiple datasets provide compelling evidence of the GLCS framework's effectiveness. By encouraging models to focus on causally relevant features and implement nuanced group-level constraints, we have shown that it is possible to develop machine learning systems that are both more equitable and more robust. The implications of this research are particularly significant for high-stakes decision-making domains such as healthcare, finance, and criminal justice, where algorithmic fairness is paramount. Our work provides a practical pathway toward developing automated systems that can handle complex intersectional data distributions more reliably and ethically.

Future research directions include extending the GLCS framework to additional domains, exploring more sophisticated cost-sensitive optimization techniques, and developing more comprehensive metrics to evaluate group fairness and group robustness. As machine learning continues to play an increasingly critical role in societal decision-making, methodologies like GLCS will be crucial in ensuring that these systems remain both performant and fundamentally fair.

References

Barocas, S., Hardt, M., Narayanan, A. (2023). *Fairness and machine learning: Limitations and opportunities*, MIT press.

Bertsekas, D. P. (1976). Multiplier methods: A survey, *Automatica* **12**(2), 133–145.

Borkan, D., Dixon, L., Li, J., Sorensen, J., Thain, N., Vasserman, L. (2019). Limitations of pinned auc for measuring unintended bias, *arXiv preprint arXiv:1903.02088* .

Borkan, D., Dixon, L., Sorensen, J., Thain, N., Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification, *Companion proceedings of the 2019 world wide web conference*, pp. 491–500.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss, *Advances in neural information processing systems* **32**.

Caton, S., Haas, C. (2024). Fairness in machine learning: A survey, *ACM Computing Surveys* **56**(7), 1–38.

Chouldechova, A., Roth, A. (2020). A snapshot of the frontiers of fairness in machine learning, *Communications of the ACM* **63**(5), 82–89.

Chuang, C.-Y., Mroueh, Y. (2021). Fair mixup: Fairness via interpolation, *arXiv preprint arXiv:2103.06503* .

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., Belongie, S. (2019). Class-balanced loss based on effective number of samples, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277.

Dablain, D., Krawczyk, B., Chawla, N. (2022). Towards a holistic view of bias in machine learning: Bridging algorithmic fairness and imbalanced learning, *arXiv preprint arXiv:2207.06084* .

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* .

Du, M., Yang, F., Zou, N., Hu, X. (2020). Fairness in deep learning: A computational perspective, *IEEE Intelligent Systems* **36**(4), 25–34.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R. (2012). Fairness through awareness, *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226.

Flach, P., Kull, M. (2015). Precision-recall-gain curves: Pr analysis done right, *Advances in neural information processing systems* **28**.

Freeman, E. A., Moisen, G. G. (2008). A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa, *Ecological modelling* **217**(1-2), 48–58.

Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q. (2017). On calibration of modern neural networks, *International conference on machine learning*, PMLR, pp. 1321–1330.

Han, X., Chi, J., Chen, Y., Wang, Q., Zhao, H., Zou, N., Hu, X. (2024). FFB: A Fair Fairness Benchmark for In-Processing Group Fairness Methods, *Proceedings of the International Conference on Learning Representations*.
https://openreview.net/forum?id=TzAJbTClAz

Hardt, M., Price, E., Srebro, N. (2016). Equality of opportunity in supervised learning, *Advances in neural information processing systems* **29**.

Hashimoto, T., Srivastava, M., Namkoong, H., Liang, P. (2018). Fairness without demographics in repeated loss minimization, *International Conference on Machine Learning*, PMLR, pp. 1929–1938.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hernández-Orallo, J., Flach, P., Ferri Ramírez, C. (2012). A unified view of performance metrics: Translating threshold choice into expected classification loss, *Journal of Machine Learning Research* **13**, 2813–2869.

Hirzel, M., Ram, P. (n.d.). Oversampling to repair bias and imbalance simultaneously, *AutoML Conference 2023*.

Kazemi, H. R., Khalili-Damghani, K., Sadi-Nezhad, S. (2023). Estimation of optimum thresholds for binary classification using genetic algorithm: An application to solve a credit scoring problem, *Expert Systems* **40**(3), e13203.

Kearns, M., Neel, S., Roth, A., Wu, Z. S. (2018). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, *International conference on machine learning*, PMLR, pp. 2564–2572.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data, *IEEE transactions on neural networks and learning systems* **29**(8), 3573–3587.

Kirichenko, P., Izmailov, P., Wilson, A. G. (2022). Last layer re-training is sufficient for robustness to spurious correlations, *arXiv preprint arXiv:2204.02937* .

Kleinberg, J., Mullainathan, S., Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores, *arXiv preprint arXiv:1609.05807* .

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I. et al. (2021). Wilds: A benchmark of in-the-wild distribution shifts, *International conference on machine learning*, PMLR, pp. 5637–5664.

Koyejo, O. O., Natarajan, N., Ravikumar, P. K., Dhillon, I. S. (2014). Consistent binary classification with generalized performance metrics, *Advances in neural information processing systems* **27**.

LaBonte, T., Muthukumar, V., Kumar, A. (2024). Towards last-layer retraining for group robustness with fewer annotations, *Advances in Neural Information Processing Systems* **36**.

Lipton, Z. C., Elkan, C., Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize f1 measure, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, Springer, pp. 225–239.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., Finn, C. (2021). Just train twice: Improving group robustness without training group information, *International Conference on Machine Learning*, PMLR, pp. 6781–6792.

Liu, Z., Luo, P., Wang, X., Tang, X. (2015). Deep learning face attributes in the wild, *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A. (2021). A survey on bias and fairness in machine learning, *ACM computing surveys (CSUR)* **54**(6), 1–35.

Qiu, S., Potapczynski, A., Izmailov, P., Wilson, A. G. (2023). Simple and fast group robustness by automatic feature reweighting, *International Conference on Machine Learning*, PMLR, pp. 28448–28467.

Robles, E., Zaidouni, F., Mavromoustaki, A., Refael, P. (2020). Threshold optimization in multiple binary classifiers for extreme rare events using predicted positive data., *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*.

Sagawa, S., Koh, P. W., Hashimoto, T. B., Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, *arXiv preprint arXiv:1911.08731* .

Sanchez, I. E. (2016). Optimal threshold estimation for binary classifiers using game theory, *F1000Research* **5**.

Sangalli, S., Erdil, E., Hötker, A., Donati, O., Konukoglu, E. (2021). Constrained optimization to train neural networks on critical and under-represented classes, *Advances in neural information processing systems* **34**, 25400–25411.

Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., Vertesi, J. (2019). Fairness and abstraction in sociotechnical systems, *Proceedings of the conference on fairness, accountability, and transparency*, pp. 59–68.

Shui, C., Xu, G., Chen, Q., Li, J., Ling, C. X., Arbel, T., Wang, B., Gagné, C. (2022). On learning fairness and accuracy on multiple subgroups, *Advances in Neural Information Processing Systems* **35**, 34121–34135.

Subramanian, S., Rahimi, A., Baldwin, T., Cohn, T., Frermann, L. (2021). Fairness-aware class imbalanced learning, *arXiv preprint arXiv:2109.10444* .

Sulaiman, M., Roy, K. et al. (2024). The fairness stitch: A novel approach for neural network debiasing, *Acta Informatica Pragensia* **13**(3), 359–373.

Tarzanagh, D. A., Hou, B., Tong, B., Long, Q., Shen, L. (2023). Fairness-aware class imbalanced learning on multiple subgroups, *Uncertainty in Artificial Intelligence*, PMLR, pp. 2123–2133.

Vapnik, V. (1991). Principles of risk minimization for learning theory, *Advances in neural information processing systems* **4**.

Wan, M., Zha, D., Liu, N., Zou, N. (2023). In-processing modeling techniques for machine learning fairness: A survey, *ACM Transactions on Knowledge Discovery from Data* **17**(3), 1–27.

Yan, S., Kao, H.-t., Ferrara, E. (2020). Fair class balancing: Enhancing model fairness without observing sensitive attributes, *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1715–1724.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification, *Journal of Machine Learning Research* **20**(75), 1–42.

Zhang, Z., Song, Y., Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5810–5818.

Zhou, S., Zhang, Y. (2016). Active learning for cost-sensitive classification using logistic regression model, *2016 IEEE international conference on big data analysis (ICBDA)*, IEEE, pp. 1–4.

Zhou, Z.-H., Liu, X.-Y. (2005). Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on knowledge and data engineering* **18**(1), 63–77.

Received December 4, 2024 , accepted January 26, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 129-156
https://doi.org/10.22364/bjmc.2025.13.1.07

Conceptual Model: Personal Information Management Using Adaptive Information Systems

[1]Boriss MISNEVS, [2]Sergejs PASKOVSKIS

[1]Transport and Telecommunication Institute, 2 Lauvas Str., Riga, LV-1019, Latvia
[2]VYBAG AB, Mossabäcksvägen 1, Stehag, Sweden, 241 74

bfm@tsi.lv, sergikpas@gmail.com

ORCID 0000-0002-3311-6507, ORCID 0009-009-3436-1289

Abstract. In the digital age, information overload is a widespread issue that hampers the efficient management of large volumes of personal data. Considering solving this problem, this study aims to develop and test a conceptual model of an adaptive personal information management system (AIS). The information system can be adapted to suit individual users' preferences and tasks. The study uses a mixed methods approach, combining quantitative and qualitative data collected through expert assessments using the Delphi method, modeling scenarios, and literature reviews. The AIS framework, validated by experts, focuses on centralizing, classifying, prioritizing, and filtering information based on user behavior and preferences. Simulations indicate that the AIS can reduce cognitive load and enhances information processing efficiency. This paper discusses the implications of these findings and offers recommendations for adaptive systems integration in personal information management.

Keywords: Adaptive Information Systems, Information Overload, Personal Information Management, Context-Awareness.

1. Introduction

Personal information management (PIM) in today's digital environment faces major challenges due to the exponential growth of information and data sources. People frequently feel overloaded with information, which impairs their ability to be productive and make wise decisions. Managing large amounts of data, storing personal information in different formats across numerous devices, managing various kinds of personal information, and projecting the future value of personal information are some of these challenges (Jones, 2017). Individuals put a lot of time and energy into organizing their data, but they frequently find it challenging and ineffective (Oh, 2019).

Due to human uniqueness, all people have different information preferences, backgrounds, levels of education, methods for processing information, and cognitive capacities (Arnold, 2023). Humans are unique and have individual information demand and processing capabilities. These requirements raise a significant concern about information systems' inability to address information overload with a "one approach fits all" solution.

Adaptive Information Systems is a relatively new research area that lies at the intersection of Information Science, Human-Computer Interaction, and Artificial Intelligence. It offers an alternative to the traditional "one-size-fits-all" approach in Information Systems development. These systems create a model based on each user's goals, preferences, and knowledge, and use this model throughout the interaction to tailor the system's responses to the user's specific needs (Palm, 2020).

The current study introduces a conceptual model of an adaptive personal information management system. This model incorporates various user characteristics that contribute to calculating the information overload ratio. Based on this ratio, the system triggers actions aimed at reducing information overload for the user. By focusing the measurement of information overload, the system adopts a proactive approach to enhancing the user's information management experience. The study also demonstrates simulation results for two scenarios: when user experience information overload and when AIS triggers actions to mitigate it

2. Purpose of the Study and Research Design

The primary aim of this study is to develop a conceptual model of AIS that effectively manages information overload by adapting to individual user characteristics. The research addresses the following hypotheses and questions:

Hypothesis: An Adaptive Information System that considers individual user characteristics will reduce user information overload.

Research Questions:

1. How can AIS identify information overload?
2. What strategies can be used to reduce information overload?

Research Design: The study employs a mixed-methods approach, combining qualitative and quantitative research methods. It involves:

1. Literature review to identify factors contributing to information overload.
2. Design and development of the AIS model.
3. Expert assessment using the Delphi method.
4. Simulation scenarios to validate the model.

3. Literature Review

Existing research highlights the adverse effects of information overload on cognitive performance and stress levels. Studies by Baskerville (2011) and DesAutels (2011) underscore the necessity for personalized information management solutions. However, modern systems lack adaptive mechanisms for dynamically adjusting user behavior and preferences. This research gap necessitates the exploration of AIS, which integrates principles from Information Science, Human-Computer Interaction, and Artificial Intelligence to offer a tailored user experience.

3.1. PIMS

In 2011 Richard Baskerville in the article "Individual information systems as a research arena" described the phenomenon named Personal Information Management System. He identifies a significant lacuna in the scientific literature that fails to adequately describe how individuals interact with information and information systems. This oversight is notable, as most scholarly attention is devoted to complex organizational systems, often neglecting the nuanced ways in which individuals engage with information systems in their daily lives (Baskerville, 2011).

Baskerville questions the overarching emphasis on organizational systems within research, urging a reevaluation of the roles personal systems play. He argues that individuals are not merely passive consumers or recreational users of technology; rather, they actively manage, curate, and utilize information to fulfill diverse personal and professional roles. This active engagement with information extends beyond traditional computing environments to include interactions with a broader digital ecosystem, encompassing social media, the Internet of Things, and beyond.

Warraich (2018) and Hwang (2015) further elaborate on the capabilities and significance of PIMS, noting that these systems are not only about managing and storing traditional data like documents and emails but also about seamlessly integrating user data across diverse platforms. PIMS employ advanced technologies such as semantic search layers and personal ontologies to enhance the meaningfulness of data and support sophisticated data analysis and mining capabilities. These functions facilitate the creation of a personalized information environment where tasks and information are synchronized across devices and platforms, enhancing both user connectivity and insight.

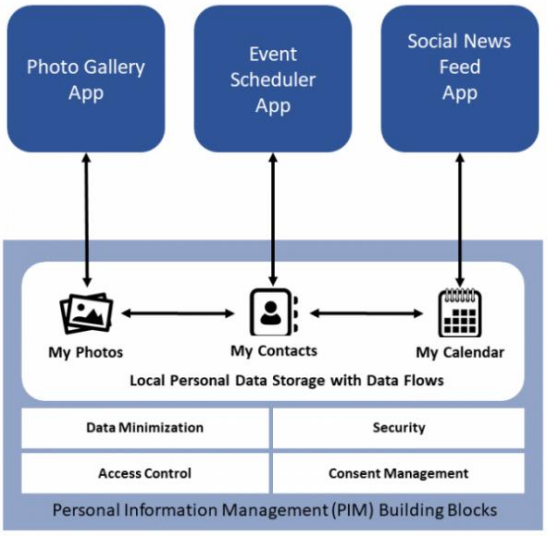


Figure 1. Schema for a Personal Information Management System with a local personal data storage (Attoresi, 2020).

However, it is critical to note that the idiosyncratic nature of PIM practices means they vary significantly across individuals, shaped by specific personal needs and environmental affordances (Warraich, 2018). This variability presents a challenge for designing universally effective PIMS, as the "one-size-fits-all" approach is often inadequate. Therefore, further research is necessary to understand the diverse ways in which different users interact with PIMS and to develop adaptive systems that can cater to this wide range of user behaviors and preferences.

By shifting the research focus from predominantly organizational information systems to include individual-oriented studies, scholars can better understand and enhance the efficacy of personal information management in our increasingly digitized lives. This approach will also allow for a more nuanced understanding of how personal information management systems can act as integral components of modern digital infrastructures, effectively bridging the gap between individual needs and technological capabilities.

3.2. Information management organization process

Variety of information about people's lives and activities is included in the definition of personal information. Jones (Jones, 2007) states that there are several ways in which one may use the term "personal information," such as information that an individual retains for their own use, information about an individual maintained by and controlled by third parties, and information that an individual experiences but is not in control of. It's important to keep in mind that personal information can come in a variety of formats, including paper and electronic documents, emails, web references, handwritten notes, materials related to work, music, videos, photos, financial information, and records of various interactions, including phone conversations, emails, and web browsing history (Jones, 2007, Widjaja, 2019). It also emphasizes how sensitive information may vary depending on the kind of data; among the most sensitive categories of data include financial information, identity, and personal papers. Widjaja suggests that while utilizing cloud storage services, consumers may perceive varying levels of privacy and security dangers connected with different categories of personal information (Widjaja, 2019).

Information organization includes the ability to gather, store, organize, preserve, retrieve, use, and share personal information in an efficient manner. This process involves several actions and steps, such as identification, comparison and examination, creation and modification, initialization, and temporary classification. These phases illustrate the steps, ideas, choices, and elements involved in a comprehensive organization of personal data. Those who keep things organized can boost output, make information easier to access, get reminders for activity management, and help people learn the material better. This procedure is required to guarantee that people safeguard their private data and maximize their time, money, and energy (Oh, 2023).

Jones has introduced term "personal information space" which refers to a person's domain that contains personal information items over which the person has at least a minimal amount of control. Books, paper documents, e-mail correspondence, e-docs, and other files are included in this area, along with links to websites and programs that facilitate the gathering, storing, retrieving, and use of information (Jones, 2007). In addition, the personal information space includes structures that help with information management, including folders and related characteristics (Jones, 2007).

The PIS model consists of a person's unique personal space of information, incorporating a range of personal information items that are, at least nominally, under the person's control. This model delineates the actions, thoughts, and decisions involved in the organization process and identifies factors that impact the process, offering valuable insights into personal information organization (Oh, 2022).

Three primary components comprise Jones' concept:

- Keeping: Input tasks that help create and preserve a personal information space are among the primary tasks carried out by personal information management systems.
- Finding: Extracting data from private areas.
- M-level activities: The term "M-level activities" describes the information-related managerial tasks that people perform. These tasks need more comprehensive and strategic measures linked to efficiently managing and arranging personal data. This involves tasks including keeping data organized, preserving privacy and security, managing information systems, assessing the efficacy of information management plans, and deciphering personal data.

These activities are central to managing personal information effectively and efficiently within the PIMS framework (Kearns, 2014). It is crucial to consider how these elements are connected to task loading and human-computer interaction in order to give a summary of the "Money, Energy, Time, and Adjustment" (META) principle (Szalma, 2007). According to the principle, in contexts involving human-computer interaction, these four factors are critical in determining task performance effectiveness and stress levels. Each component has an impact on how users interact with interactive tasks and how they perform as a result. People can improve their interactions with technology, effectively manage cognitive loads, reduce stress, and improve task performance and user satisfaction by striking a balance between resources like money, energy, time, and adjustment.

3.3. Decision making in personal information management

An individual's decision-making processes can be greatly impacted by how people arrange and manage their personal information. Having easy access to pertinent information, lowering cognitive load, and promoting more informed decisions are all ways that well-organized information can improve decision-making (Bergman, 2008). Stress levels, cognitive load, task complexity, and individual capabilities are the factors that influence on decision making and are essential components to human-computer interaction and influence how decisions are made (Szalma, 2007).

3.4. Issues and challenges in personal information management

There are several challenges for humans while working with PIM: such as information fragmentation, challenges locating and retrieving information, information value volatility, difficulties dealing with the sheer volume of information, and obsolescence of technology. Web-based information, email, and desktop computers are just a few of the crucial domains that have identified as requiring information management (Jones, 2007)

The study highlighted several issues related to information management across various demographics, underscoring not just information overload but also "filter failure" as

critical concerns (Kearns, 2014). Challenges include the difficulty of managing abundant online information (Majid, 2010), fragmentation across devices leading to inefficiencies and data loss (Majid, 2010), and the emotional and practical burdens of managing health-related information (Sannon, 2023). Additionally, security and privacy concerns are particularly pronounced among young adults using internet services (Majid, 2010). The study also noted significant dissatisfaction with current methods for handling household information (Sannon, 2023).

People often experience frustration and time loss when they struggle to find important information quickly, which can lead to feelings of confusion, helplessness, and irritation. Fragmented data, spread across multiple devices or tools, intensifies issues of information overload, reducing productivity and efficiency (Chaudhry, 2015). Additionally, the increasing diversity of information formats and technologies complicates information management, leading to mistakes and poor decisions about data storage that hinder information retrieval (Jones, 2007).

3.5. Information overload

When people believe they are getting too much information, it is impeding their ability to complete tasks, a negative psychological state known as information overload occurs. Emotional and cognitive difficulties are indicators of information overload, which is most likely caused by intrinsic and extrinsic information characteristics, poorly defined information needs, the work environment, or the information environment. Internal and external consequences arise from the emotional and cognitive manifestations. Poor decision-making can have a negative impact on finances and human resources(Bellabes, 2022).

Ineffective information management practices lead to information overload, which in turn impairs information retrieval. The ease with which vast amounts of information can be accessed via a variety of devices exacerbates this issue. Every new technological advancement contributes to the already-existing information overload by making information production, acquisition, and dissemination even simpler and less expensive (Majid, 2010). When staff members don't have enough time to read, comprehend, and make use of the information that is available, information overload happens. When attempts to locate information are unsuccessful, this leads to frustration. This may result in anxiety, which can then lead to headaches, dizziness, disorientation, helplessness, and irritation and annoyance.

Excessive amounts of potentially helpful and relevant information can impede rather than assist people in their tasks, according to descriptions of the phenomenon (Bawden, 2020). Information diversity, complexity, choices, confusion, and harm are all relevant factors to consider in addition to its quantity (Bawden, 2020). The idea of information overload has also been linked to the connection between an abundance of information and a sense of psychological overload (Mostak, 2014). Comprehending the complex characteristics of information overload entails taking into account both the subjective reaction of each individual to the information as well as its objective quantity and quality (Mostak, 2014). Another aspect the concept of "information noise" embodies a diverse array of perspectives and interpretations, reflecting the multifaceted nature of the term (Spira, 2012). It's associated with the perception of excessive or irrelevant data, which hinders the efficient processing and utilization of pertinent information. The impact of

information noise on individuals and organizations can be profound, contributing to factors such as inefficiency, misinformation, and reduced decision-making capacity (Bawden, 2020). In general, information overload is problematic because it has been associated with diminished reasoning ability and decision quality, poorer memory recall and feelings of confusion, stress, and anxiety (Schick, Gordon & Haka, 1990). Various attention deficit problems are thought to be associated with information overload Rose, 2010).

3.6. Notification's disruption

Several studies demonstrated that the number of incoming messages can in fact considerably add to information overload (Kumar, Shrivastav, 2013). A deluge of information may result from the ease with which information is shared and accessed, particularly with the use of digital workplace tools and various communication platforms. This can make it difficult for people to sort through and handle the volume of messages (Kumar, 2013).

Lowered notification-related disruptions may ultimately result in increased productivity, according to empirical data. They are reported that, there is a possibility that with fewer notifications interfering with work, individuals may perform better and become less irritated within the day. These disruptions are viewed as stressors that prevent people from meeting objectives on important tasks, jeopardizing daily targets and causing anxiety (Ohly, Bastin, 2023).

3.7. Consequences of information overload

Information overload, cognitive load, security worries, and privacy issues are just a few of the drawbacks of digital information processing. Numerous studies and literature sources demonstrate these impacts (Sannon, 2023). For example, emphasizes how handling digital information can be cognitively taxing, making decision-making and retrieval difficult. Additionally, it has been observed that the existence of digital information overload raises the risk of security breaches, anxiety, and stress (Mahler, 2020, Majid, 2010).

Poor organization and difficulties with predictability resulting from mishandling personal digital information can cause a decrease in productivity and an increase in stress (Jones, 2007, Warraich, 2018). Focusing too much on digital document management, especially with ineffective tools or techniques, can take a lot of time and mental energy, which can lead to frustration and decreased productivity (Jones, 2007). As explained by Sanon (Sanon, 2023), the primary underlying causes of information overload and cognitive load include things like being overexposed to information, the speed at which technology is developing, and the incapacity to effectively handle and process information (Mahler, 2020). Decision-making, mental workload, and job satisfaction suffer because of these underlying causes, which also lead to increased information processing, shorter task completion deadlines, and more pertinent information than can be processed (Mahler, 2020).

3.8. Mitigating information overload

Information overload won't affect those who prefer to receive little to no information but are presented with a lot of it because they won't process it all. There are several strategies to mitigate information overload, including instituting email policies to limit excessive communication, storing knowledge in repositories, and filtering and arranging information flow. In addition to addressing concerns about the quantity and caliber of information shared within the organization, these actions aim to lessen the burden that information overload is putting on individuals who are experiencing it (Mahler, 2020). The prevention and recovery strategies that are examined to lessen cognitive fatigue and information overload include a range of approaches. Taking notes, sticking notes on the wall, and organizing paperwork are examples of preventative measures that one can use to ward against potential forgetfulness and serve as a reminder of helpful tools (Elsweiler, 2007).

To lessen the cognitive load associated with information management, techniques like active document triage and the use of passive information management tools like auto filtering and tagging options (Sanon, 2023).

The effectiveness of PIM can be influenced by an individual's cognitive processes, which include their mental strategies for information processing, organization, and retrieval (Kearns, 2014). Furthermore, knowing one's preferred ways of thinking and learning can help create customized PIM strategies.

As result create comprehensive and individualized information management strategies, it is crucial to consider cognitive, behavioral, and technological aspects of PIM when analyzing individual factors.

The authors do not exclude using other pedagogical methods that can help to develop students' critical thinking, such as flipped classrooms (Vdovinskiene, 2023). Creating an individualized learning experience that meets each student's unique needs, preferences, and learning speed is a key component of adaptive learning to user behavior. Using user models to automate the learning process, this method uses adaptive learning management systems, which modify learning paths and content delivery in real-time in response to assessments of a learner's engagement and level of knowledge. Through constant observation and adjustment to user conduct, these systems seek to improve student performance and guarantee that the educational process is as efficient and pertinent as feasible, ultimately cultivating a more customized and encouraging learning environment (Jurenoka and Grundspenkis, 2023).

3.9. Adaptive information systems

Adaptive Information Systema is a management information system adapting its user interface and interaction strategy depending on user preferences and past user behavior and satisfaction (Höpken, 2018). Based on perceived changes in the environment, system conditions, and requirements, it entails altering its structure, parameters, and behavior at run time. The system's ability to adapt lets it function well in a variety of situations. To learn more about the concept of "behavioral adaptation," we can consult pertinent information in (Shuetz, 2020), a document that addresses the adaptability of Cognitive Computing Systems (CCS). Behavioral adaptation refers to the ability of CCS to adjust their actions over time to achieve improved outcomes. By utilizing adaptive features, this

unique capability allows for the creation of more effective systems that meet user preferences and needs more effectively.

The system should be made to be able to adapt to changing environments without breaking down in order to maximize human performance within an adaptive system. When we talk about the adaptive aspect of AIS, we mean the system's capacity to adjust to external stimuli while maintaining its functionality. The capacity for adaptation enables any system, be it artificial or biological, to adjust efficiently to changing circumstances and increase overall performance (Kovacs, 2004).

Understanding the mechanisms through which adaptation takes place is essential to understanding how adaptivity functions in adaptive systems. Adaptive systems use a variety of techniques for self-adaptation, such as keeping an eye on their surroundings and changing course when necessary. One strategy makes use of online reinforcement learning, in which the system gains knowledge about the efficacy of adaptation actions through real-time interactions with its surroundings (Palm, 2020). By having the system learn and modify its behavior based on predetermined learning goals, this approach improves the system's adaptability without requiring manual intervention, thereby automating the task of developing self-adaptation logic. Instead of requiring information system engineers to manually create self-adaptation logic, systems can now learn it automatically through the integration of online reinforcement learning. This method lets the system learn and adapt in real-time through machine learning, automating the laborious engineering task of creating self-adaptation logic (Palm, 2020).

Additionally, in the context of human-machine interaction, the cognitive effects of prolonged continuous use require adaptive interfaces that can recognize impaired cognitive states and modify the interaction to maintain efficiency and safety (Palm, 2020). These mental state-based adaptive systems use a variety of metrics to determine the operator's current mental state, then adjust the interaction accordingly. Changes in task scheduling, information presentation, or stimulus salience may all be part of the adaptation, which is intended to counteract performance decline brought on by extended interaction.

3.10. Research gap

The development and comprehension of technologies that can centralize information flow and dynamically adapt to user behavior and preferences remain largely unexplored, despite the existence of research in personal information management and personal information management systems.

Current PIM systems manage information in a fragmented way, with different tools and apps dealing with emails, calendars, tasks, and social media. Research is needed on technologies that can bring these disparate information sources together into one cohesive system. Centralizing information flow can make management easier, eliminate duplication, and offer a complete overview of all user activities.

Although there may be some degree of customization available with current systems, there aren't many cutting-edge adaptive technologies that can automatically modify and optimize information management procedures based on user interactions and behaviors. Research ought to concentrate on creating adaptive PIM systems that dynamically

customize the user experience based on real-time behavior and preferences by utilizing artificial intelligence and machine learning.

System awareness and adaptability to various user contexts are necessary for effective PIM (e.g. work, private, and social). Investigating the potential of contextual data, including time, location, device usage, and user activity, to customize information management tactics requires more research. By minimizing cognitive load and preserving productivity, context-aware PIM systems enable users to move between contexts with ease.

4. AIS Conceptual Model Development

The AIS conceptual model was designed, and data was collected using a combination of qualitative and quantitative research methods. This mixed-approaches approach guaranteed a comprehensive understanding of the problems associated with information overload and the efficacy of the AIS. The requirements for the conceptual model were initially established and later verified by professionals. Expert suggestions were also incorporated into the design of the conceptual model. The study ends with simulations that explore the scenario of information overload and AIS impact on it.

The initial conceptual model of the AIS developed based on a review of existing studies and theories in personal information management, information overload, and adaptive systems. This model aimed to address key challenges in managing information overload by incorporating adaptive mechanisms, user-centric design principles.

To validate the conceptual model, the Delphi method was employed to get expert opinions. Experts were selected based on their extensive knowledge and experience in information technologies. In the Delphi method were included closed-ended questions to gather quantitative data on the frequency of experiencing information overload and the perceived efficacy of features offered by the AIS. Among the expert responses, these data were utilized to find patterns and trends.

The simulation environment for assessing the Adaptive Information System (AIS) was constructed using AnyLogic software, which enabled detailed modeling of the AIS and its components. Two primary scenarios were devised to mirror real-world conditions and evaluate the system's impact: the first scenario, "Without AIS," served as a baseline for measuring the typical extent of information overload, while the second scenario, "With AIS," was used to evaluate the effectiveness of the AIS in significantly reducing information overload.

To simplify the management of information overload, the model introduces the concept of an Activity. An activity represents an information item, (an email, note, or task etc.) and includes a number the actions required to process this information item. By referring to information items as activities, the AIS can better organize, structure and manage personal information. The number of activities collectively creates the total volume of information that the user must process. The Context is specified, which indicates the environment or situation in which the activity occurs. It helps to understand the background or setting of the activity. The actions are connected sequentially, indicating the order in which they must be performed.

Figure 3 shows how activities and the actions that go along with them add up to the total amount of information that a user must process. This helps to visually represent the idea of information load within the AIS for Personal Information Management.

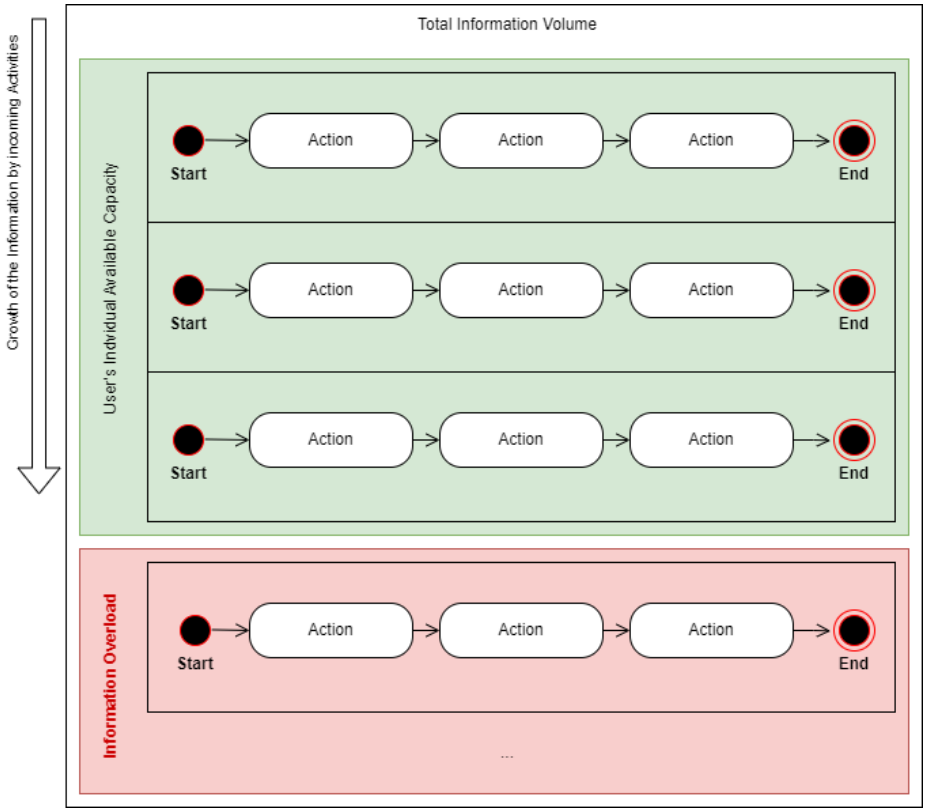


Figure 2. Information in context of AIS

At the top, the diagram introduces the concept of Information as the top element being managed. This information is broken down into a Collection of Activities, each representing a specific information item, such as an email, note, or task. The Individual Capacity axis on the left side of the diagram represents the user's cognitive ability to process information. This capacity influences how many activities and actions a user can handle without experiencing information overload. The Total Information Volume axis on the right side of the diagram indicates the cumulative amount of information generated by the activities. The number of activities and the complexity of their actions collectively determine the total volume of information a user needs to manage. All activities and their corresponding actions add up to the total information volume that the user needs to process.

In the study, was chosen scenario to demonstrate how the AIS handles information overload and simulate a situation in which a user is distracted by a new activities and switching their attention on it and delaying other ongoing tasks.

In this scenario, the user is involved in multiple activities with different contexts (e.g. work and private).

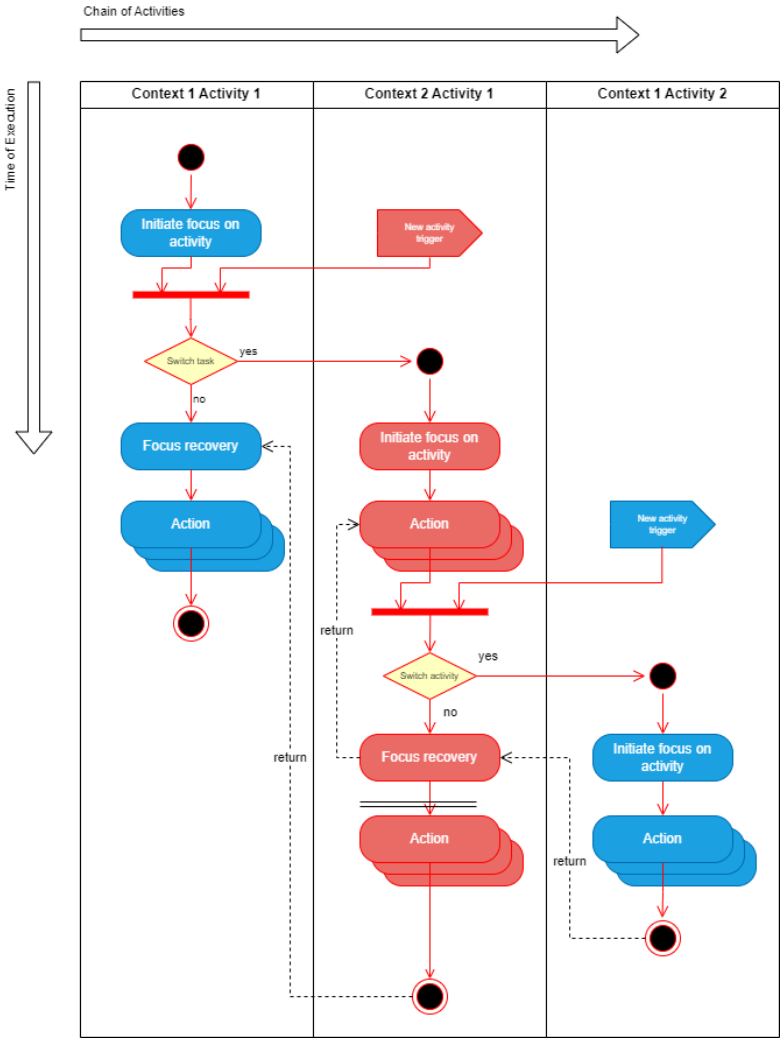


Figure 3. Distracting the user and switching between activities

The flowchart (see Fig. 4) illustrates the sequence of activities and decisions involved when a user is interrupted by a new activity. The user begins by initiating focus on Activity 1 within Context 1. This involves starting the sequence of actions required to complete the task.

When new activity (Context 2, Activity 1) is triggered and interrupts the current task. The user decides whether to move to the new activity based on priority and context. If the new activity is deemed important, the user switches to Context 2, Activity 1. The user initiates focus on Activity 1 in Context 2, temporarily leaves the previous task, and performs the actions required by Activity 1 in Context 2. After completing the previous task, the user returns to Context 1, Activity 1.

User interference can lead to information overload, forcing the user to switch to another activity. When the user returns to the previous activity, it can take up to 22 minutes to fully engage with the task again, according to recovery time research (Mark et al., 2008). Additionally, frequent glitches negatively impact task completion and contribute to delays, further increasing the user's cognitive load and reducing overall productivity.

Classifying new activities by priority and determining whether the user should switch tasks, the AIS can efficiently manage interruptions. This promotes maintaining productivity and dealing with distractions. Simulation demonstrates the AIS's ability to manage information overload by prioritizing tasks, minimizing unnecessary context switches, and incorporating focus recovery periods.

Simulation without AIS regulation

The main objective of this scenario is to evaluate the baseline performance of the Personal Information Management System with adaptive features disabled. In this scenario, the system functions normally and without adaptive intervention. Incoming data is not subject to filtering. The user receives all information as he receives it, without any prioritization or classification. This scenario is intended to illustrate the difficulties and problems that often arise in non-adaptive systems, including possible interference from frequent and unfiltered notifications and information overload. Metrics to watch include the start and completion time of user activity.

Simulation with AIS regulation

The goal of this scenario is to demonstrate how the AIS works and the benefits it provides when its adaptive features are enabled. The focus is on how the AIS can reduce information overload through intelligent interventions. In this scenario, the adaptive system - including information filtering - works as intended. The system dynamically adjusts the criteria based on the user's preferences and current tasks, as well as preset criteria for filtering incoming data

4.1. Calculating information overload

In order to effectively manage personal information and prevent information overload, the AIS requires a reliable method to calculate and assess the user's information load. In this study a simplified equation was developed based on Jackson formula (Jackson, 2012) and derived from the concept of activities described previously. The first step in calculating information overload involves determining the capacity required for each activity. An activity is comprised of multiple actions, each with its own processing time. The total required capacity for an activity is calculated as follows (see Formula 1):

$$Activity\ Required\ Capacity = \sum_{i=1}^{n} Action\ Processing\ Time_i \quad (1)$$

Once the required capacity for each activity is determined, the overall information overload can be assessed by comparing the total required capacity of all activities to the user's available capacity (see Formula 2):

$$Information\ Overload = Available\ Capacity - \sum_{i=1}^{n} Activity\ Required\ Capacity_i \quad (2)$$

In this context:

- If the value of Information Overload is < 0, it indicates that the user is experiencing information overload.
- If the value of Information >= 0, it indicates that the user is not experiencing information overload.

This equation considers the total required capacity for all activities and compares it to the available capacity of the user. The AIS calculates the metrics of every action and adjusts them according to user behavior to determine if the user has enough time to process the information.

4.2. Adaptive System Framework

The conceptual model self-regulation component was designed using the MAPE-K framework, which is proposed for use in self-regulation systems. MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) represents a conceptual framework for building self-adaptive systems that can adjust behavior in response to changes in the environment or the system itself. The MAPE-K loop provides a structured approach for designing systems, emphasizing the importance of a continuous feedback loop supported by a knowledge base (Iglesia, 2015).

4.3. Additional functionality considerations

The model should ensure seamless cross-platform functionality, allowing consistent access and management of information across various devices through a unified network. It should integrate diverse sources of personal information into a single platform and utilize advanced search and retrieval technologies like natural language processing. The system needs automated organization features and robust privacy and security measures, alongside scalability to handle growing data and user numbers. Additionally, it should support collaboration and secure sharing, continuously adapt to user preferences, and employ strategies to mitigate information overload, thereby enhancing user experience and efficiency.

4.4. Expert review process

A diverse group of experts was selected for their academic and professional expertise in personal information management, adaptive systems, and user experience design. The panel consisted of 11 experts. The experience levels of the experts varied, providing a well-rounded view of the issues related to information overload varies from <5 years to

more than 20 years. In total participated 11 experts. 9 of the experts have bachelor's degree and 2 – masters.

All of them are actively practicing professionals, with most working in information technology and professional in marketing and sales.

4.5. Draft questionnaire

The questionnaire used in the study focused on several key areas to gather expert insights into the design and effectiveness of AIS. First, it addressed experts' personal experiences with information overload, asking them to describe the frequency and sources of such overload and the distractions they encounter. This helped to understand the real-world impact of information overload on professionals.

Second, the questionnaire explored the impact of these distractions on productivity and information management, probing how these interruptions affect daily work and efficiency. Third, it sought expert opinions on the potential of AIS to enhance productivity by reducing distractions and adapting to the unique preferences of individual users.

Additionally, experts were asked to identify crucial features and functionalities that an AIS should have to manage and reduce information overload effectively. They also discussed methods for measuring information complexity and overload, as well as strategies to mitigate these challenges effectively.

The questionnaire included both closed and open questions, allowing for structured responses and more detailed explanations. Feedback from these rounds was summarized and shared in subsequent rounds to refine the questions further and clarify any ambiguous areas, ensuring a comprehensive understanding of expert perspectives on AIS.

5. Simulation setup and execution

The simulation is performed using AnyLogic software and it was configured to model the AIS and its components: activities, actions and dynamic adaptation strategies.

5.1. Simulation

To determine the necessary number of simulations runs, a statistical formula was used to achieve a 95% confidence level with an 80% estimated proportion of observing information overload, and a 5% margin of error. This formula initially suggested 246 runs. However, given the practical considerations of each simulation's 14-day duration, authors adjusted the number to 25 runs per scenario (both with and without AIS enabled) to balance thoroughness and computational feasibility.

5.2. Synthetic data generation

For this study, synthetic data was created to mimic the environment and evaluate the effectiveness of the AIS. An Italian software company provided a data set that served as the basis for the synthetic data creation process and gave the simulation scenarios a realistic basis. The purpose of the synthetic data is to reproduce realistic user behavior and activity patterns, including typical distributions of activity types, frequencies, durations,

and complexity levels. To ensure that the data adequately reflects typical user patterns of information management and information overload, additional insights and parameters from previous studies were included. The synthetic data set was created by applying random sampling techniques to these distributions.

5.3. Simulation data analysis and validation

To evaluate the AIS's impact on information overload, collected data was analyzed focusing on key performance metrics such as overload reduction, task completion rates, and user satisfaction. Statistical tests, specifically the McNemar test, were used to ascertain the significance of differences between scenarios with and without AIS intervention.

The analysis included six types of activities—Emails, News, Mobile Notifications, Task Assignment, Task Resolution, Task Charge—categorized into 'personal' and 'work' contexts, with a typical distribution of 25% personal and 75% work activities per day. Daily activities were set from 8:00 to 17:00, not exceeding 125 activities to simulate a realistic workday environment, with tasks generated from real-world data to inform the simulations.

Detailed Breakdown of Tasks

The creation of task activities is based on the analysis of dataset for one of its users. This real-world data provides a foundational basis for simulating work-related tasks and activities within the AIS model.

- Task Assignment:
  - Number of activities per day: 1
  - Duration: 1.14 hours (min: 0.0003 hours, max: 10.71 hours)
- Task Resolution:
  - Number of activities per day: 1
  - Duration: 3.47 hours (min: 0.0003 hours, max: 10.90 hours)
- Task Charge:
  - Number of activities per day: 2
  - Duration: 2.66 hours (min: 0.0008 hours, max: 11.17 hours)

Additional metrics in the study include title and content sizes, with titles averaging 75 words and emails ranging from 59 to 206 words, based on recommendations for optimal engagement. Newspaper content was set at 516 words. The recovery time after an interruption is standardized at 25 minutes. Task criticality varies with 54.72% of tasks being critical and lasting seven days, 30.52% rated high with a five-day duration, and 14.76% medium, lasting two days. The metrics 'ActivityRequiredTime' and 'AvailableTime' were employed to measure information overload, which is calculated hourly using a Information Overload formula (see Formula 2).

Validation AIS impact

The effects of AIS on managing information overload are validated using the McNemar test. The McNemar test was used to statistically evaluate whether the implementation of AIS significantly enhances information overload management compared to a baseline scenario without AIS intervention.

6. Conceptual model simulation and validation

6.1. Conceptual model design

The concept of how people manage personal information in a digital environment was explained by Baskerville and Gass (Baskerville, 2011, Gass, 2015), and this understanding served as the basis for the conceptual model design requirements. User-centered design, context support, cross-platform functionality, privacy and security, advanced search and query, automated organization, advanced search and query, and scalability are considered when designing the conceptual model.

6.2. Conceptual model framework

The conceptual framework for AIS emphasizes the centralization of both incoming and outgoing information. This centralization is critical to effective and efficient personal data management as it streamlines processes and reduces complexity for the user. The AIS acts as a central hub and enables better organization and handling of information, helping to mitigate problems such as information overload.

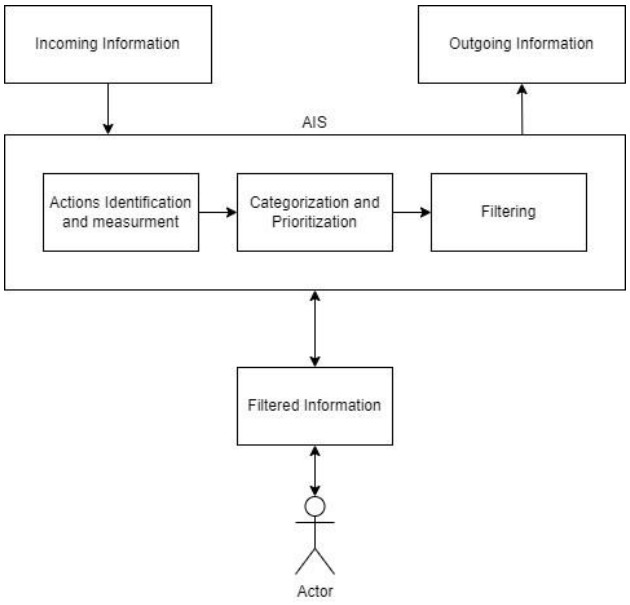


Figure 4. AIS framework

The AIS acts as a central hub (see Fig. 5) for all incoming and outgoing information. Incoming information includes all data coming from various sources such as emails, notifications, documents and social media updates. In order for this raw data to be useful

to the user, it must be processed and managed. The AIS processes this information through several key components:

Action identification and measurement: This module identifies necessary actions based on the incoming information and measures relevant parameters such as urgency, relevance, and context. This step is crucial for understanding the importance and priority of each piece of information.

Categorization and prioritization: After identifying actions, the system organizes them into categories and assigns priority levels. This ensures that the most critical information is addressed first, helping the user to manage tasks effectively and efficiently.

Filtering: The filtering engine removes unnecessary or irrelevant information, ensuring that only the most relevant data reaches the user. This process includes relevance filtering, which eliminates information that does not meet the user's current needs, and spam filtering, which removes low-quality content.

After processing, the filtered information is presented to the user. This refined data is tailored to the user's preferences and needs and represents actionable and relevant information with which the user can interact. The user, referred to in this context as the "actor", accesses this filtered information via the AIS.

Outbound information generated by the user's interactions with the AIS includes responses, new tasks, and actions taken based on the filtered information. This outgoing information is also managed by the AIS, which can further process or disseminate it as necessary, maintaining a coherent and organized flow of information.

Centralization within the AIS is key to better management of personal data. It simplifies the management process, reduces the time and effort required to process multiple streams of information, and ensures consistency in categorizing, prioritizing, and filtering information.

6.3. Assessment by Experts

The Delphi method was used to validate the conceptual model of the AIS. This process involved obtaining expert opinions through several rounds of questionnaires containing both closed- and open-ended questions. The aim was to refine the model and ensure that it effectively addresses the challenges of information overload. The process included three rounds of questionnaires to obtain expert opinions on the proposed AIS conceptual model. The iterative process aimed to refine the conceptual model, overcome the challenges of information overload, and ensure that the system's functions are consistent with expert recommendations. These rounds resulted in consensus on several key aspects of the AIS and provided valuable insights for its development and implementation.

Spreading information overload

Experts repeatedly reported information overload and highlighted its importance as a critical issue that needs to be addressed. Digital notifications and frequent context switching have been identified as sources of distraction, negatively impacting productivity and highlighting the need for effective management strategies.

Support for adaptive systems

There was a strong consensus that an AIS had the potential to increase productivity by reducing distractions and effectively managing information. Experts emphasized the

importance of an AIS that adapts to individual user behavior, preferences and context changes and provides personalized management suggestions and configurations.

Key features and functions

Experts identified and refined several important features and functions of the AIS:

- Automated Tagging: Widely supported (91%), automated tagging based on content analysis was seen as beneficial for enhancing information retrieval and organization.
- Integration with Existing Tools: Prioritized integration with personal email clients (100%), calendar applications (82%), and task management tools (45%) to streamline information management.
- Delegating decisions to an AIS scheduling and calendar management (91%), email filtering and responses (82%), social media management (54.5%), task prioritization (100%).
- AI-Driven Prioritization: Using AI to prioritize information based on urgency and context was recommended to manage high-priority information effectively.
- Customizable Notifications: Suppressing notifications during focus times, summarizing information at intervals, and allowing user-customizable notification settings were highlighted as effective strategies for minimizing distractions.

Privacy and security

The importance of implementing robust data protection and security measures was strongly emphasized. Experts emphasized the need for encryption, strict access controls, regular audits and transparency about data usage to ensure user trust and protect sensitive information.

User control and convenience through automation

Retaining user control over automated decisions was deemed crucial. Experts preferred complete control over system decisions with the ability to override and customize actions. While there was some variability in comfort levels, the majority were at least somewhat comfortable with the AIS making routine or low-risk decisions.

The consensus across all three rounds highlights the importance of addressing information overload with an adaptive system that includes key features such as task prioritization, content grouping, automated tagging, integration with existing tools, AI-driven prioritization and customizable notifications. Robust security measures and user control over automated decisions are also important components. These insights provide a solid foundation for the AIS conceptual model and guide its development to effectively meet user needs and increase productivity.

6.4. Conceptual model

The conceptual model diagram of AIS in Personal Information Management represents various components and their interactions intended to enable efficient management and retrieval of information (see Fig. 6).

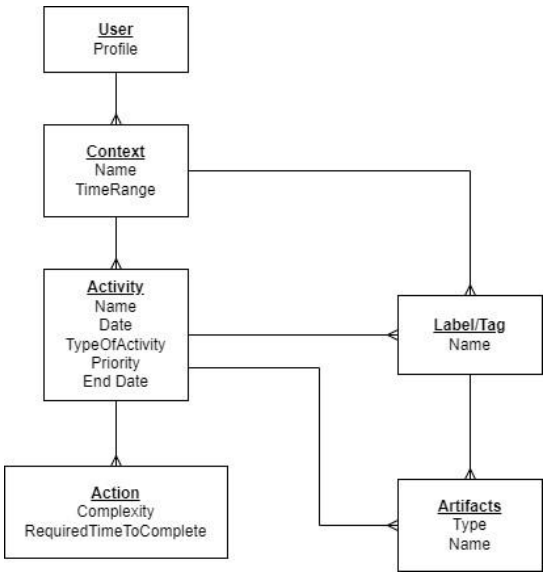


Figure 5. Conceptual model diagram

User: At the center of this model is the user, characterized by a profile that contains user-specific information. This profile influences the different contexts in which the user operates.

Contexts are defined by their name and time range, which indicate the specific situation or environment (e.g. work or personal) and the time period in which the context is relevant. Contexts are directly linked to the user, indicating that different users may have different contexts based on their profiles. Activities take place within these specific contexts and show the relationship between the context and the actions taken by the user.

An activity is nested in contexts and is linked to labels/tags and actions. Actions, on the other hand, represent the steps or tasks required to complete the activity. Each action is characterized by its complexity and RequiredTimeToComplete, which estimate the difficulty and time required to complete the task.

Labels/Tags are essential for organizing activities and artifacts. This categorization makes it easy to locate activities and artifacts and ensures that users can quickly find the information they need.

Artifacts represent different types of documents or files associated with activities. Each artifact is defined by its type (e.g. a document or an image) and its name. Artifacts are linked to both activities and labels/tags. This dual mapping means that certain documents or files are relevant to specific activities and are categorized for easy access.

6.5. Simulation development

The simulation process models the adaptive behavior of the AIS and shows how it interacts with user activities, categorizes and prioritizes tasks, and automates decision making when

necessary. This simulation provides insight into the effectiveness of AIS in mitigating information overload and streamlining information management tasks.

General simulation parameters

User reading speed – parameter define user individual characteristics, that impacts on speed how fast user can read and process content. This criteria stronger participate in activity process review and simulates user reading behavior. In the model has been used average human reading speed – 238 (Rayner et al., 2016) words per minute, by modifying it will impact on the AIS activities planning.

Activity type – in the simulation consist of 4 activity types:

- News feed – defines all information that relates and incoming as a news.
- Mobile notification – activity describes any type of information that comes to user as notification like: new email, application notifications etc.
- Task – this activity describes user main task, that is coming as a regular work. In the simulation considered user tasks based on dataset analysis.
- Email – described user incoming emails.

All activities except tasks distributed between work and personal context in ratio – 75% activities about the work and 25% - personal. Tasks have been created only in work context.

Task activities - distributed across 3 types:

- Assign ticket seriousness – user review ticket and apply criticality on the ticket
- Resolve the ticket – user review and close the task as accomplished
- Take in charge ticket – when user actively working on the ticket.

Tasks distributed according to the dataset analysis and should be assign average a 1 ticket per day, and can be minimum 1 and maximum 12 tickets per day. Then should be 1 resolved ticket per day with 1 minimum number and 10 max and “take in charge ticket” average 2 per day and min 1 and maximum 16 tickets.

Every task has been distributed according to criticality. There has been identified that tickets have 3 types of ticket priority. Based on the analysis those has been distributed accordingly (see Table 1):

Table 1.

Tickets criticality distribution

Criticality name	Distribution	Task duration (days)
Normal	0.5472	7
Important	0.3052	5
Critical	0.1476	2

In the simulation, users engage in both work-related and personal activities, which include emails, news, and notifications. Users receive an average of 120 activities daily: 15 news items, 64 notifications, and about 5 tasks, with the latter consuming roughly 7 hours of the day, leaving approximately one hour for reading emails and news. Activity distribution is as follows: news feeds under 15%, emails at 55%, and notifications at 40%.

The simulation spans 14 continuous workdays from 8:00 AM to 5:00 PM, distributing tasks primarily from 8:00 AM to 2:00 PM, peaking at 9:00 AM.

Simulation flowchart

The simulation process begins with the arrival of an activity that represents an information item or a task that requires the user's attention. The AIS assesses whether it is capable and able to handle the activity autonomously. When AIS is enabled, the activity is routed to AIS where the activity can be immediately processed and placed in a queue or deferred to the user for manual processing. The simulation workflow captures these interactions and illustrates the logical flow of activities through the AIS-enabled system.

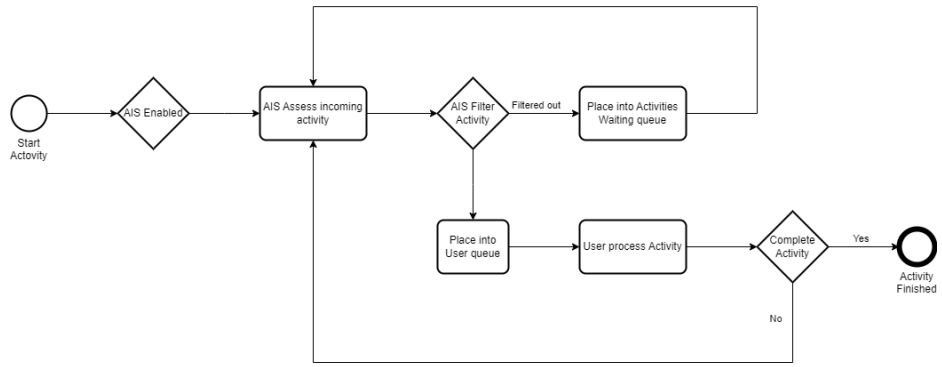


Figure 6. Simulation flowchart

The flowchart (see Figure 6) details the workflow of the AIS simulated in Anylogic. When a new Activity the system first checks if AIS is enabled. If so, AIS processes the task by categorizing and prioritizing it according to preset rules and preferences. Tasks not immediately processed are moved to a AIS waiting queue, until they can be addressed to the user. The user interacts with tasks in the processing stage, and if a task isn't completed— due to an interruption—it's cycled back to the queue. Successfully completed tasks are then marked as finished and removed from the workflow.

The Activity agent in the simulation represents the various tasks and information items that the user needs to process. Each Activity agent contains multiple parameters that define its characteristics, and the actions required for its completion. This section describes the parameters and their roles in simulating the dynamics of information management within the AIS.

Activity

In the simulation, activities were defined by multiple parameters to reflect realistic scenarios. Context distinguishes between "work" and "personal" activities, influencing prioritization. Activities contained information about device whether the activity originated from a laptop or smartphone. Importance, categorized as low, normal, or high, is based on predefined expert rules.

Every activity has a number of actions parameter, that specifies the complexity of activities, ranging from 1 to 5 actions, while RemainingTimeList and RemainingTime

track the time required to complete actions and tasks. Activities are timestamped with start date and end date to aid scheduling, and artifacts (e.g., attachments) influence action complexity.

The simulation also includes parameters for interruptions to track context switching and recovery.

User behavior modeling

The user agent in the simulation mimics user interactions with activities. It starts in an Idle state, where no activities are engaged, and transitions to a Ready state when prepared to accept new tasks. If there are pending activities in the queue, it moves to Review Activity state, where it may simulate a device or context switch, introducing delays ranging from 30 to 180 seconds to represent distractions from incoming notifications. Additional delays of 30 to 60 seconds simulate internal distractions when switching between applications.

If interrupted, resuming to a previous activity might take 900 to 1200 seconds for the user to reacquaint themselves with the task, based on prior research. Following the review, the user continue to working on Activity. If an ongoing activity is of lower priority, it is swapped for a new, higher-priority activity. Within this state, the user processes necessary actions to complete the activity. Once completed, the user entered the Free state, indicating readiness to tackle the next activity or return to waiting for new tasks.

Simulation execution

The current simulation models the daily activities of an office worker over a 14-day period with working hours from 8:00 till 17:00. Each working day, the user receives between 121 and 133 activities, including emails and notifications. This range is based on research by Vengagge (2023) and Acer (2015) and reflects the typical information load of office workers. The simulation considered that users receive activities in proportion to 25% personal activities and then 75% work-related activities.

To accurately represent the variety of information found, activities are categorized into two contexts: work-related and personal. This distinction simulates the different types of activities users might be exposed to throughout the day and provides a comprehensive overview of professional and personal information management.

Activity generation schedule

The simulation is designed to mirror the natural distribution of activities throughout a typical day. Work-related activities are concentrated during standard office hours, from 8:00 a.m. to 5:00 p.m., reflecting real-world professional communication patterns. Conversely, personal activities are scheduled primarily outside of work hours, continuing into the evening. This scheduling ensures that the simulation realistically captures the flow of work and personal tasks. The activities are delivered to users through devices such as laptops and smartphones. In the simulation model, users can process only one activity at a time, maintaining a focused workflow that simulates real-life task management.

Logic of the AIS Regulation

AIS uses a rule-based approach to filter and prioritize tasks in the simulation. While advanced techniques like dynamic classification are not implemented, activities are assigned predefined importance labels that guide their prioritization. If an activity lacks a scheduled start date, the system calculates one by subtracting the estimated time required

to complete the task from its deadline. This ensures that tasks are processed within their specified timeframe.

Priority Calculation

The AIS assigns priorities to tasks based on their urgency and scheduled dates. For tasks with future start and end dates, they are added to a waiting list, provided the list contains no more than 50 activities. If a task is due today or its execution is overdue, its priority increases proportionally to the delay, ensuring that critical tasks are addressed promptly. Overdue activities are marked as "disturbed," signaling their critical status and need for immediate attention. This prioritization system enables the AIS to manage tasks effectively, focusing on the most urgent and important activities.

Device Context and Activity Status

To minimize cognitive load, all activities are handled within a single device context, ensuring users operate solely on their laptops. This approach avoids the distraction of switching between devices, streamlining task management. Once priorities are calculated, the AIS updates the status of each activity and processes them according to urgency and relevance. This mechanism allows the system to dynamically adapt to changing workloads, minimizing delays and enhancing productivity.

6.6. Data analysis and validation

The efficacy of the Adaptive Information System (AIS) was rigorously tested through a comprehensive simulation campaign using AnyLogic software version 8.9.0. A total of 50 simulation runs were conducted, equally split between scenarios with AIS enabled (25 runs) and disabled (25 runs), generating nearly equal activities in each state—47,189 activities with AIS disabled and 47,197 with it enabled, totaling 92,513 activities across all simulations.

The simulations were designed to mimic real-world workflows, distributing activities between personal (25%) and work-related (75%) tasks to reflect typical usage scenarios. The activity distribution was meticulously tracked, with personal activities comprising 4,670 emails, 2,320 news items, and 4,643 mobile notifications, while work-related activities included more complex tasks such as 221 task assignments, 455 task resolutions, and 655 task charges.

To simulate a typical day’s information flow, activities were generated on a schedule from 8 AM to 5 PM, peaking during midday hours. For example, from 8-9 AM, 3 activities were generated, increasing to 9 by 10 AM, and peaking at 25 activities between 12-1 PM, before tapering off in the late afternoon.

Moreover, the nature and intensity of activities varied, with each type of activity requiring a different number of actions, influenced by the complexity of the task. Notably, tasks were programmed to require more actions due to their longer average duration, aligning with the protocol that divided the total execution time of tasks by an 8-minute interval per action. Overall, the average time spent on activity actions was approximately 18,268.82 seconds (about 5 hours) per day.

Extending the validation, several simulation scenarios were tested over a 30-day period to ensure the AIS's robustness and reliability over extended operational periods. These extended simulations confirmed that the AIS rules functioned as expected, without any

degradation in performance, and all required activity parameters were consistently met across varied metrics.

Critically, the effectiveness of the AIS in managing information overload was quantitatively assessed using the McNemar test, comparing the binary responses of overload occurrence between AIS-enabled and disabled states. The results were striking:

With AIS Enabled: 0 occurrence of overload.

With AIS Disabled: 5,730 occurrences recorded.

This statistical analysis showed a dramatic reduction in information overload cases with the activation of AIS, substantiated by a chi-squared statistic of 5729.8 and a negligible p-value, indicating a highly significant improvement in managing information flow and reducing overload.

Comprehensive data validation underscores that the AIS not only performs effectively under typical usage conditions but also maintains its efficacy in extended and varied operational scenarios. This robust performance highlights the system's potential to significantly enhance productivity and decision-making by efficiently managing and prioritizing tasks, thereby reducing the cognitive load on users.

7. Results

In the simulations, both scenarios (with and without AIS) were executed with a similar total number of activities, with 47189 and 47197 respectively. However, the difference in the outcomes of these activities is significant. With AIS intervention, nearly all task activities that are started are also finished, with 0 delayed out of 6528 activities being completed per round. In stark contrast, without AIS intervention of the 6528 task activities delayed (see Table 2).

Table 2.

Simulation results

AIS status	Created activities	Finished activities	Interrupted activities	Delayed activities
Disabled AIS	47189	24159	47189	6528
Enabled AIS	47197	26817	0	0

One of the most striking differences between the two scenarios is the number of interruptions. The AIS effectively eliminates interruptions during activity processing, with zero interruptions recorded per round. Conversely, without AIS, there are a significant number of interruptions, averaging 2109 per round. This highlights the AIS's capability to provide a more focused and uninterrupted workflow for the user.

Moreover, the AIS reduces the number of delayed activities. There were no activities that were delayed per round with AIS intervention, whereas, without AIS, there are 6528 delayed activities. This indicates that the AIS not only helps in managing activities more efficiently but also ensures that they are completed on time.

The AIS significantly improves task completion rates and reduces interruptions, highlighting its effectiveness in managing information overload and enhancing user productivity. There were 5730 observations of information overload recorded prior to AIS disable, but there were none when AIS was turned on. These results demonstrate that the AIS significantly enhances efficiency by ensuring that almost all started activities are completed with minimal delays and no interruptions. This leads to higher productivity as the AIS reduces the cognitive load on the user, minimizes delays, and eliminates interruptions, allowing for better time management and task performance.

8. Discussion

The design of the conceptual model is grounded in understanding how people manage personal information in a digital environment, incorporating user-centered design, context support, cross-platform functionality, privacy and security, advanced search and retrieval, automated organization, and scalability. The central element of the conceptual model of the AIS is the centralization of incoming and outgoing information, which optimizes processes and reduces complexity for the user.

The AIS conceptual model includes several components: the user profile, contexts, activities, labels/tags, and artifacts. The user profile influences various contexts, in which activities occur, while labels/tags and artifacts help organize and retrieve information efficiently. This model emphasizes how users interact with contexts, manage activities, perform actions, and utilize artifacts, highlighting the importance of labels/tags in organizing information.

The Delphi method was implemented to validate the AIS conceptual model through expert assessments. The consensus across all three validation rounds underscores the importance of addressing information overload with an adaptive system. It incorporates automated tagging, integration with existing tools, AI-driven prioritization, and customizable notifications.

The simulation development process in AnyLogic models the adaptive behavior of the AIS, demonstrating its interactions with user activities, task categorization, prioritization, and decision-making automation. Activities are processed through a workflow that includes decision points for AIS processing, waiting queues, and user interactions. The simulation results highlight the effectiveness of the AIS in managing activities compared to the scenario without AIS intervention.

The findings validate the hypothesis that AIS reduces information overload. The system's ability to adapt to user behavior and preferences is crucial in managing cognitive load. The research questions are addressed through the AIS model development and validation, to provide actionable insights into effective strategies for reducing information overload.

9. Conclusion

This study presents a novel, validated conceptual model for an AIS that effectively addresses the pressing challenge of information overload in the digital age. The basis of the AIS is its ability to adapt dynamically to individual user preferences and contexts,

providing a personalized approach to information management. The study demonstrates the system's potential to significantly reduce cognitive load and improve performance using a robust multi-method approach that combines expert review and modeling. The AIS aims to simplify personal information management to improve decision-making, reduce stress, and boost productivity in both personal and professional contexts. Therefore, AIS in today's digital environment where information overload is a common problem will be a useful tool. Future research should focus on implementing AIS in the real world, examining how it integrates with current digital systems, and assessing its long-term impact on user performance and behavior. Exploring the integration of more advanced machine learning algorithms is also highly necessary. Implementing these improvements will enable the system to anticipate users' information needs and adapt to changes, improving overall system management.

Further research should include quantitative analyses demonstrating the impact of AIS on cognitive load and performance, as well as user feedback and case studies illustrating real-world implementations and user experiences. This will help to strengthen the AIS's validity and applicability. To give the AIS's personalized nature, ethical and privacy concerns must be addressed, to guarantee that the system strictly maintains user confidentiality and data protection guidelines.

By making progress in these areas, the AIS model has the potential to transform personal information management and establish new standards for adaptive digital technologies in the future.

Acknowledgments

The authors extend their deepest gratitude to all university colleagues whose invaluable assistance and unwavering support greatly contributed to the completion of this research article.

A heartfelt thanks are extended to Professor Irina Yatskiva for her exceptional support in the research methodology.

References

Arnold, M., Goldschmitt, M., Rigotti, T. (2023) Dealing with information overload: a comprehensive review, *Frontiers in Psychology*. 14:1122200, 1-28.

Attoresi, M., Moraes, T., Zerdick, T. (2020). EDPS TechDispatch : personal information management systems. Issue 3, 2020.

Bawden, D., Robinson, L. (2020). Information Overload: An Introduction, in *Oxford Research Encyclopedia of Politics*. Oxford University Press.

Baskerville, R. (2011). Individual information systems as a research arena. *European Journal of Information Systems*, **20**(3), 251–254.

Bergman, O., Beyth-Marom, R., Nachmias, R. (2008) The user-subjective approach to personal information management systems design: Evidence and implementations, *Journal of the American Society for Information Science and Technology*, **59**(2). 235–246.

Bruder, J. (2022). The Algorithms of Mindfulness. *Science Technology and Human Values*, **47**(2), 291–313.

DesAutels, P. (2011). UGIS: Understanding the nature of user-generated information systems, *Business Horizons*, **54**(3), 185–192.

Gregory, M., Descubes, I. (2011). Personal Knowledge Management: PIMS/IIS/UGIS A Research in Progress, in *International Days of Statistics and Economics, Prague, September 22-23, 2011*, 159–168.

Hwang, Y., Kettinger, W.J., Yi, M.Y. (2015). Personal information management effectiveness of knowledge workers: Conceptual development and empirical validation, *European Journal of Information Systems*, **24**(6), 588–606.

Jackson, T. W., Farzaneh, P. (2012). Theory-based model of factors affecting information overload. *International Journal of Information Management*, ***32***(6), 523–532.

Jones, W., Dinneen, J.D., Capra, R., Diekema, A., Pérez-Quiñones, M. (2017). Personal Information Management, in *Encyclopedia of Library and Information Science, Fourth Edition*. CRC Press, pp. 3584–3605.

Jurenoka, S., Grundspenkis, J. (2023). Development of Methods and Models for Generating an Adaptive Learning Plan Based on the User's Level of Knowledge, *Baltic Journal of Modern Computing*, **11**(1), 90-113.

Mähler, V. (2016). Too much information!: Information-overload from an IT-management perspective. diva-portal.org

Mark, G., Gonzalez, V.M., Harris, J. (2005). No task left behind? in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, pp. 321–330.

Oh, K. E. (2019). Personal information organization in everyday life: modeling the process. *Journal of Documentation*, **75**(3), 667–691.

Ohly, S., Bastin, L. (2023). Effects of task interruptions caused by notifications from communication applications on strain and performance. *Journal of Occupational Health*, **65**(1), e12408.

Palm, A., Metzger, A., Pohl, K. (2020). Online Reinforcement Learning for Self-adaptive Information Systems, in *International conference on advanced information systems engineering, 2020,* Springer, pp. 169–184.

Vdovinskienė, S. (2023). Using Flipped Classroom as an Active Teaching Method for Teaching Engineering Graphics, *Baltic Journal of Modern Computing*, **11**(3), 383-397.

Warraich, N.F., Ali, I., Yasmeen, S. (2018). Keeping found things found: Challenges and usefulness of personal information management among academicians, *Information and Learning Science*, **119**(12), pp. 712–720.

Widjaja, A.E., Chen, J.V., Sukoco, B.M., Ha, Q.A. (2019). Understanding users' willingness to put their personal information on the personal cloud-based storage applications: An empirical study, *Computers in Human Behavior*, **91**, pp. 167–185.

Received September 10, 2024, revised January 30, 2025, accepted February 18, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 157–165
http://doi.org/10.22364/bjmc.2025.13.1.08

A Qualitative Comparison of the State-of-the-Art Next-Best-View Planners for 3D Scanning

Andrejs ARISTOVS[1], Evalds URTANS[2]

[1] Riga Technical University, Riga, Latvia
[2] Riga Technical University, Department of Artificial Intelligence and Systems Engineering, Riga, Latvia

andrejs.aristovs@edu.rtu.lv, evalds.urtans@rtu.lv

ORCID 0009-0008-1806-1772, ORCID 0000-0001-9813-0548

Abstract. This is a survey paper in which we review the state-of-the-art Next-Best-View planners with the focus on their application in solving an autonomous 3D scanning task. According to market reports, the 3D scanning market will continue to grow in response to the increasing demand for augmented and virtual reality solutions. Taking into account that the number of skilled 3D artists is limited and their labor is highly paid, an alternative way of creating high quality 3D models is 3D scanning existing objects. In many cases, 3D scanning is the only way to get photorealistic textures and high-definition models. Automated 3D scanning can be used as a way to preserve art, document changes in the environment, create detailed models of consumer products. Six next-best-view planners were compared using ROS in the Gazebo simulation environment. The MA-SCVP machine learning method achieved on average 93.1% coverage, that is 5.9% higher than ScanRL, 36% higher than SEE, and 1% higher than volumetric information gain methods. Maximum coverage with the MA-SCVP method was achieved after 12.2 views on average, versus 20 views for the volumetric information gain methods.

Keywords: Next-best view, 3D reconstruction, ROS, Machine Learning

1 Introduction

It is estimated that the 3D scanning market will reach a billion dollars by 2024 as discussed in Kari (2022), the main reason being the applications of AR and VR mostly in the marketing field. According to Boland (2020), photorealistic models create a sense of craving in consumers, improve conversion, and increase session length. In return an improvement in this metrics results in higher income and growth in customer satisfaction. In many AR/VR applications, photorealistic 3D models improve immersion and blend better in the scene. Such models can be created either by skilled 3D artist or by means of automated 3D scanning of real-world objects. So far only large companies

such as IKEA have had the ability to digitize their products and create 3D assets of their inventories.

By using appropriate 3D scanning techniques, it is possible to democratize the creation of 3D models. In our experience, structured light scanning yields the best results, by controlling lighting, polarization and camera focus, high detail models with HDR textures can be achieved. The main hurdle is an intelligent way to plan the path with the intent of reducing the necessary number of view points. Structured light scanning creates high-resolution textured 3D models, but each new scan takes up to 10 seconds, depending on the number of patterns being projected. Most of this time is to make sure the system is static, dampen the vibrations, and adjust camera lenses focal length and aperture. With this in mind, we evaluated state-of-the-art NBV planners with the focus on minimal view coverage.

2 Related work

Depending on the target use for the NBV planner different metrics are used to evaluate planner's performance. As mentioned earlier for a system that uses structured light 3D scanning, most important metric is number of views needed to achieve threshold reconstruction quality.

2.1 Next-best view planner comparison techniques

Authors are aware of the last analysis of NBV planning methods carried out by Scott et al. (2003), in which the comparison metrics have been defined to evaluate different approaches. The evaluation criteria of this publication were reviewed as a basis for our literature analysis.

Many of the state-of-the-art NBV planners are iterations and improvements of previous methods, as the MA-SCVP method introduced in Pan et al. (2023) is an improvement of the SCVP method introduced in Pan et al. (2022) that additionally uses PC-NBV by Zeng et al. (2020) neural network to define the best view. NBV-Net 4-5 neural network architecture introduced in Vasquez-Gomez et al. (2021) (the numbers in the name stand for: 4 convolutional layers and 5 fully connected layers), is based on the previous paper Mendoza et al. (2019) NBV-Net network architecture that contained 3 convolutional layers and 5 fully connected layers. The authors also tested other NBV-Net configuration, like NBV-Net 3-3, NBV-Net 3-5, NBV-Net 4-3, and NBV-Net 5-4, with the conclusion that the NBV-Net 4-5 network achieves the best results. NBV-Net was the first 3D convolutional network architecture applied to solving 3D reconstruction. Multiple further solutions use a similar network architecture and use NBV-Net as ground truth.

An alternative to deep machine learning (ML) based methods are measurement direct methods (SEE Border et al. (2018) and SEE+ Border and Gammell (2022)), (PC-NBV Zeng et al. (2020)) - all using point cloud data to define the region of interest and the next best view.

After literature analysis, the most prominent NBV planners have been selected for further evaluation: MA-SCVP Pan et al. (2023), SEE Border and Gammell (2022),

ScanRL Peralta et al. (2020) and volumetric information gain methods such as AE (Average Entropy), RSE (Rear Side Entropy), RSV (Rear Side Voxel), OA (Occlusion Aware) and PC (Proximity Count) by Delmerico et al. (2018) and UV (Unobserved Voxel) by Vasquez-Gomez et al. (2014) and volumetric information gain method defined in Kriegel et al. (2015).

2.2 3D model datasets

The ABC dataset introduced in Koch et al. (2018) contains more than 1 million CAD models, downloaded from the Onshape[3] platform. Shapenet Chang et al. (2015) dataset contains three millions of CAD models, 220 000 of which are categorized into 3135 classes. Thingi10K Zhou and Jacobson (2016) dataset contains 10 000 models intended for 3D printing. In general, large-scale datasets are used for training and testing ML algorithms. For 3D reconstruction tasks, smaller datasets with textured models created by 3D scanning are more common.

The models of the bunny, introduced in Turk and Levoy (1994), the dragon by Curless and Levoy (1996) and the armadillo and the Buddha by Krishnamurthy and Levoy (1996) are available on the Stanford University computer graphics laboratory website.[4] The 20 models introduced in Rodolà et al. (2013) are available on the Munich Technical University (TUM) computer vision group website.[5] The 80 detailed models created by means of structured light 3D scanning by Jensen et al. (2014) are available on the Image Analysis and Computer Graphics at the Technical University of Denmark (DTU) website.[6] Several models are available on the MIT Computer Science and Artificial Intelligence laboratory website.[7] The House3K dataset used in Peralta et al. (2020), which contains 3000 building models with textures, is available on the GitHub repository.[8] The Linemod dataset[9] used in Hinterstoißer et al. (2012) and the HomebrewedDB dataset[10] introduced in Kaskman et al. (2019) are available on the TUM websites.

3 Methodology

In order to evaluate different NBV planners, the Gazebo simulation environment was chosen, with foresight to later use NBV planners with ROS. Primarily due to authors familiarity with ROS as well as the capabilities to use both Gazebo simulation environment and real-world robotic systems. As different NBV planners were implemented in different environments, an approach only involving the view coordinates was used. By using only coordinates, multiple different environments can be utilized and the results are not software and setting dependent. The positions were transformed to be used

[3] `https://www.onshape.com/`
[4] `http://graphics.stanford.edu/data/3Dscanrep/`
[5] `https://cvg.cit.tum.de/data/datasets/clutter`
[6] `https://roboimagedata.compute.dtu.dk/`
[7] `https://people.csail.mit.edu/tmertens/textransfer/data/`
[8] `https://github.com/darylperalta/Houses3K`
[9] `https://campar.in.tum.de/Main/StefanHinterstoisser`
[10] `https://campar.in.tum.de/personal/ilic/homebreweddb/`

with Gazebo (the coordinates X, Y, Z were scaled and the rotations were converted to quaternions). The evaluation pipeline is presented in Figure 1.

The coverage percentage is calculated as a similarity between the ground-truth model and the resulting point cloud. Our approach differs from previously used methodologies as we use more comprehensive 3D test model dataset as well, compare multiple different approach performance combined with the focus on highest coverage percentage achievable in least views possible.

Algorithm 1 Similarity calculation between pointclouds

Input cloudA, cloudB, threshold
Output cloudA similarity to cloudB
$tree = open3d.geometry.KDTreeFlann(cloudA)$
$num\_outlier = 0;\ num\_valid = 0$
for pt **in** $cloudB.points$ **do**
    $dist = nearestDistance(tree, pt)$
    if $dist < threshold$ **then**
        $num\_valid = num\_valid + 1$
    else
        $num\_outlier = num\_outlier + 1$
result $= num\_valid/(num\_outlier + num\_valid)$

For the experiments, the threshold value has been set to 0.005 meters or 0.5 mm. CloudB is the point cloud of ground-truth created from the .ply model.

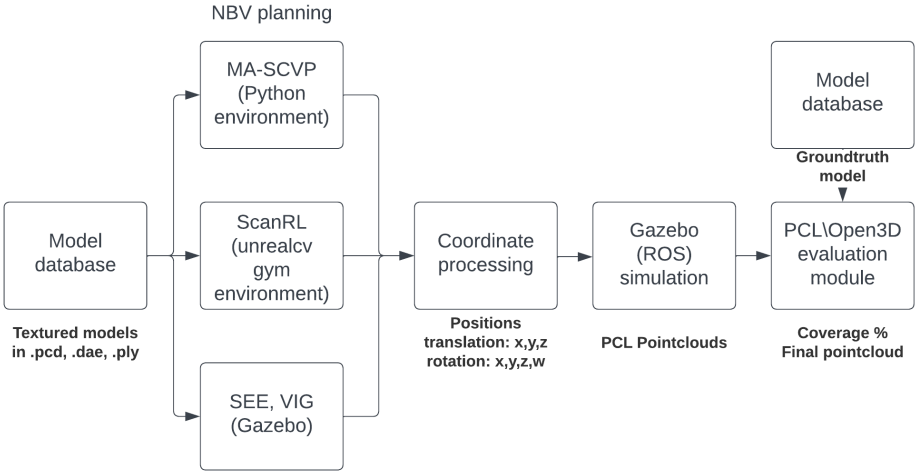


Fig. 1: NBV planner evaluation methodology

To evaluate the performance of the NBV planner, a testing dataset has been assembled. The dataset contains 9 models: bunny and dragon from the Stanford University dataset, can and cat from the LineMod dataset, and 5 models from the HomeBrewDB dataset: mug, minion, dog, stegosaur, and triceratops. Set composition was selected to include the most common test models as well as models with complicated geometries and occlusions. The models have been scaled down, and the geometry has been simplified. To improve reconstruction and perception, bright textures have been applied to the models. From the 3D models, ground truth point clouds have been created that contain on average 52 000 points. Simplified geometry models are not larger in size than 5 MB for the .ply mesh models and not larger than 12 MB for the .dae Gazebo models.

Fig. 2: Dataset of 3D models used for evaluation

4 Results

On average, with the same number of views, the MA-SCVP NBV planner achieved 9.2% higher coverage than the volumetric information gain methods. Until 7 views, the volumetric information gain metrics UV (Unobserved Voxel), AE (Average Entropy), or other volumetric information gain methods can achieve higher coverage than MA-SCVP, possibly due to MA-SCVP selecting views to optimize the local path rather than purely maximizing information gain. After about 7 views, MA-SCVP achieves higher coverage than the other evaluated methods.

MA-SCVP created on average a coverage set of 12.2 views, while ScanRL, SEE, and volumetric information gain methods were limited to 20 views. In 8 of the 9 models, MA-SCVP reached the highest overall coverage (with the exception of the bunny model, bunny the percentage of coverage shown in Figure 4). The bunny model was the only model where a larger number of views than those defined by MA-SCVP was beneficial and resulted in higher maximum coverage.

Measurement-direct approach SEE achieved on average a 36% lower maximum coverage than MA-SCVP, but gradually improved the quality of the model, where each next view increased the coverage. With volumetric information gain methods, in some cases, among the 20 views, some of the views were redundant and did not improve coverage.

Examples of the percentage of coverage after 5, 10 views and maximum achieved for the 2 selected models and some of the methods are presented in Table 1.

Table 1:

Coverage after 5, 10 views and maximum achieved.

Model	Metric	MA-SCVP	ScanRL	SEE	UV	VG	AE	RSE	OA	PC	RSV
Cat	C5	**0.874**	0.483	0.449	0.813	0.828	0.872	0.734	0.813	0.648	0.699
Cat	C10	**0.932**	0.754	0.492	0.915	0.926	0.907	0.854	0.915	0.872	0.832
Cat	Max	**0.957**	0.866	0.619	0.942	0.945	0.925	0.943	0.942	0.934	0.938
Bunny	C5	0.615	0.514	0.539	**0.787**	0.742	0.760	0.673	0.760	0.639	0.543
Bunny	C10	**0.852**	0.774	0.628	0.843	0.828	0.835	0.813	0.825	0.818	0.814
Bunny	Max	0.862	0.870	0.675	0.873	**0.895**	0.877	0.881	0.883	0.891	0.844
Dragon	R5	0.721	0.512	0.493	0.709	0.584	**0.735**	0.667	0.709	0.606	0.667
Dragon	R10	**0.857**	0.703	0.535	0.782	0.789	0.769	0.812	0.782	0.751	0.788
Dragon	Max	**0.876**	0.832	0.651	0.849	0.877	0.843	0.874	0.849	0.878	0.873
Triceratops	R5	**0.892**	0.558	0.519	0.847	0.838	0.841	0.743	0.847	0.821	0.743
Triceratops	R10	**0.948**	0.816	0.576	0.902	0.882	0.919	0.864	0.902	0.893	0.881
Triceratops	Max	**0.973**	0.900	0.597	0.952	0.956	0.946	0.956	0.952	0.953	0.953

On most of the models, coverage similar to the cat model was achieved, where MA-SCVP achieved after 5 views the highest overall coverage or within 1-2% from the maximum coverage achieved by volumetric information gain methods. It is worth mentioning that since MA-SCVP defines a minimal coverage view set, the local path is optimized. Because of this limitation, the coverage in the first views might be lower than that achieved by information gain approaches.

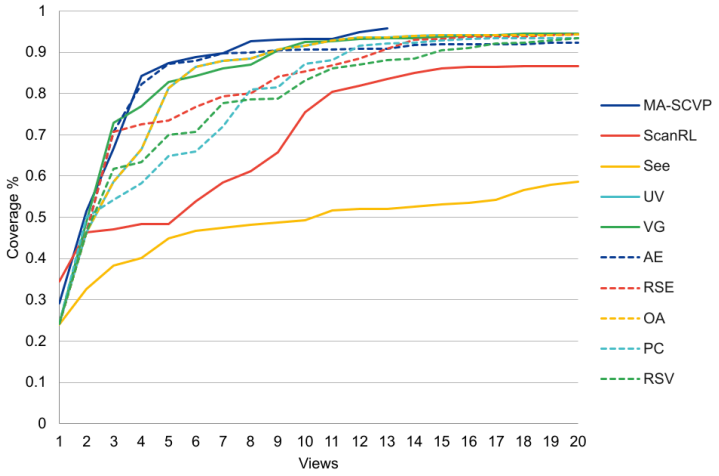


Fig. 3: Coverage for the cat model.

Th bunny model was the only model tested in which all volumetric information gain metrics as well as ScanRL achieved a higher maximum coverage than MA-SCVP. For the bunny model, MA-SCVP defined a set of 11 views, ScanRL achieved the maximum

coverage in 17 views, and the volumetric information gain methods were limited to 20 views, but did not improve more than 1% in views 17 to 20. This points to the limitation of the smallest view set defined by MA-SCVP and the value of using more views to improve coverage. The SEE in the bunny model was tested up to 50 views and achieved 74.2% maximum coverage with an average 0.9% improvement per view.

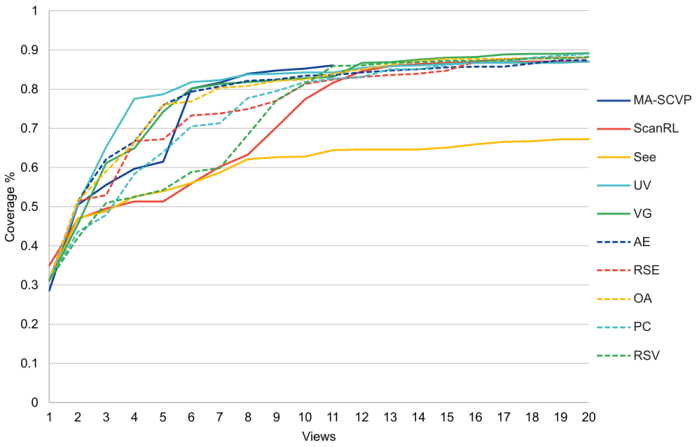


Fig. 4: Coverage for the bunny model.

5 Further research

As future work we define several possible directions.

Combining several NBV planning approaches to create hybrid NBV planners. The ML based MA-SCVP method is limited to a 32 view dataset. Combining its fast coverage in the initial views with a measurement-direct approach like SEE to define the next-best view based on the gaps in the resulting point cloud can lead to a higher overall coverage percentage.

ML model training on a larger dataset with models of higher geometric complexity. Most of the datasets do not include models with a high levels of occlusions and geometric complexity. Training ML models on a larger and more complex dataset can result in more robust NBV planners.

Neural networks with larger state and action spaces. Fixed view space is a limiting factor for neural network-based methods, as well as 32x32x32 voxel representation is not suitable for some objects, for example, plant leaves. Using higher-resolution state and action spaces might yield better results for more complex geometries.

Testing NBV planners on real-world objects with sensor sensor noise and positioning uncertainty.

6 Conclusions

In our experiments, the ML based MA-SCVP method achieved higher coverage with fewer views than other reviewed methods. By combining robotic platforms and ML NBV path planning, it is possible to optimize automated 3D asset requisition and achieve high resolution models in less amount of views.

This study emphasizes the importance of understanding the differences between NBV planners when applied to 3D reconstruction. Future research should explore hybrid methods, combine the strengths of the models discussed, and develop more adaptive strategies that can better handle a wider range of geometries.

References

Boland, M. (2020). Artillery intelligence briefing. Accessed on 01.05.2023. `https://artillry.co/wp-content/uploads/2020/08/August-2020-ARtillery-Intelligence-Briefing.pdf`

Border, R., Gammell, J. D. (2022). The Surface Edge Explorer (SEE): A measurement-direct approach to next best view planning, *The International Journal of Robotics Research (IJRR)* . Submitted, Manuscript #IJR-22-4541, arXiv:2207.13684 [cs.RO].

Border, R., Gammell, J. D., Newman, P. (2018). Surface edge explorer (see): Planning next best views directly from 3d observations, *2018 IEEE International Conference on Robotics and Automation (ICRA)* pp. 1–8.

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F. (2015). Shapenet: An information-rich 3d model repository, *ArXiv* **abs/1512.03012**.

Curless, B., Levoy, M. (1996). A volumetric method for building complex models from range images, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA, p. 303–312. `https://doi.org/10.1145/237170.237269`

Delmerico, J., Isler, S., Sabzevari, R., Scaramuzza, D. (2018). A comparison of volumetric information gain metrics for active 3d object reconstruction, *Autonomous Robots* **42**.

Hinterstoißer, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G. R., Konolige, K., Navab, N. (2012). Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes, *Asian Conference on Computer Vision*.

Jensen, R. R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H. (2014). Large scale multi-view stereopsis evaluation, *2014 IEEE Conference on Computer Vision and Pattern Recognition* pp. 406–413.

Kari, M. (2022). Augmented reality drives e-commerce growth. Accessed on 01.05.2023. `https://nordicgrowth.com/en/augmented-reality-drives-e-commerce-growth/`

Kaskman, R., Zakharov, S., Shugurov, I. S., Ilic, S. (2019). Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects, *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* pp. 2767–2776.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D. (2018). Abc: A big cad model dataset for geometric deep learning, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 9593–9603.

Kriegel, S., Rink, C., Bodenmüller, T., Suppa, M. (2015). Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects, *Journal of Real-Time Image Processing* **10**, 611–631.

Krishnamurthy, V., Levoy, M. (1996). Fitting smooth surfaces to dense polygon meshes, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA, p. 313–324. https://doi.org/10.1145/237170.237270

Mendoza, M., Vasquez-Gomez, J. I., Taud, H., Sucar, L. E., Reta, C. (2019). Supervised learning of the next-best-view for 3d object reconstruction, *Pattern Recognit. Lett.* **133**, 224–231.

Pan, S., Hu, H., Wei, H. (2022). Scvp: Learning one-shot view planning via set covering for unknown object reconstruction, *IEEE Robotics and Automation Letters* **7**, 1463–1470.

Pan, S., Hu, H., Wei, H., Dengler, N., Zaenker, T., Bennewitz, M. (2023). One-shot view planning for fast and complete unknown object reconstruction.

Peralta, D., Casimiro, J., Nilles, A. M., Aguilar, J. A., Atienza, R., Cajote, R. (2020). Next-best view policy for 3d reconstruction, *arXiv preprint arXiv:2008.12664* .

Rodolà, E., Albarelli, A., Bergamasco, F., Torsello, A. (2013). A scale independent selection process for 3d object recognition in cluttered scenes, *International Journal of Computer Vision* **102**, 129–145.

Scott, W. R., Roth, G., Rivest, J.-F. (2003). View planning for automated three-dimensional object reconstruction and inspection, *ACM Comput. Surv.* **35**(1), 64–96. https://doi.org/10.1145/641865.641868

Turk, G., Levoy, M. (1994). Zippered polygon meshes from range images, *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* .

Vasquez-Gomez, J. I., Sucar, L. E., Murrieta-Cid, R., Lopez-Damian, E. (2014). Volumetric next-best-view planning for 3d object reconstruction with positioning error, *International Journal of Advanced Robotic Systems* **11**.

Vasquez-Gomez, J. I., Troncoso, D., Becerra, I., Sucar, E., Murrieta-Cid, R. (2021). Next-best-view regression using a 3d convolutional neural network, *Machine Vision and Applications* **32**.

Zeng, R., Zhao, W., Liu, Y.-J. (2020). Pc-nbv: A point cloud based deep network for efficient next best view planning, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* pp. 7050–7057.

Zhou, Q., Jacobson, A. (2016). Thingi10k: A dataset of 10,000 3d-printing models, *arXiv preprint arXiv:1605.04797* .

Received September 7, 2023 , revised September 30, 2024, accepted January 13, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 166-176
https://doi.org/10.22364/bjmc.2025.13.1.09

Advancing Cybersecurity through AI: Insights from EU and Candidate Nations

Blerta LEKA[1], Daniel LEKA[2]

[1] Department of Mathematics and Informatics, Faculty of Economy and Agrobusiness, Agricultural University of Tirana, St. Paisi Vodica 1025, Tiranë, Albania,
[2] Special Court of First Instance for Corruption and Organized Crime, St. Jordan Misja, 1057, Tiranë, Albania

bmocka@ubt.edu.al, lekadaniel@yahoo.com

ORCID 0009-0000-6734-8237, ORCID 0009-0003-6154-9996

Abstract. Artificial Intelligence technologies are changing many sectors, but they also bring difficult challenges, especially in cybersecurity and data protection. As the adoption of artificial intelligence increases, countries face growing threats from cybercrime and must implement frameworks strong data protection. This study examines the connections between artificial intelligence, cyber security and data protection in relation to Albania and North Macedonia's progress towards integration into the European Union. The challenges and progress that these countries are experiencing to reach EU standards are assessed, especially in relation to the General Data Protection Regulation (GDPR). Using qualitative methods, including policy analysis and case studies, will help determine the effectiveness of cybersecurity and data protection frameworks.

Keywords: Digital Transformation, AI Ethics, Regulatory Compliance, Information Security, Threat Mitigation and National Cybersecurity Strategy

1. Introduction

Artificial Intelligence (AI) is a dynamic field within computer science dedicated to creating systems that can learn, reason, and operate autonomously. Russell and Norvig (2009) define AI as a focus on agents that perceive their environment and take actions based on that perception. Rich et al. (2009) further emphasize that AI's aim is to enable machines to perform tasks typically requiring human intelligence. This broad field encompasses sub-domains like deep learning, natural language processing, robotics, and computer vision. According to Gartner (2024), although over 60% of Chief Information Officers (CIOs) believe AI is essential for driving innovation, less than half are prepared to manage the associated risks.

AI is revolutionizing various industries, from healthcare and finance to transportation and telecommunications, with significant applications in cybersecurity. AI systems are particularly important in automating threat detection and anomaly identification, crucial for mitigating risks. The 2024 Cyber Security Breaches Survey highlights that a substantial number of businesses (50%) and charities (32%) in the UK faced cyberattacks

over the past year (UK Government, 2024). While larger organizations tend to implement stronger security measures, smaller entities struggle with vulnerabilities. The European Cybersecurity Strategy stresses the necessity of industry collaboration to reinforce defenses, and initiatives like Horizon Europe focus on advancing AI-driven security measures.

The European Court of Auditors (2024) points to an urgent need for a coherent regulatory framework for AI within the EU, advocating for the integration of ethical considerations while fostering innovation. The General Data Protection Regulation (GDPR) serves as the cornerstone of EU data protection law, establishing compliance standards for AI systems processing personal data to safeguard individual privacy rights (European Union). Furthermore, the AI Act aims to enhance oversight concerning AI's role in data protection, underscoring the importance of privacy by design.

The EU has been proactive in regulating AI, as evidenced by the Proposal for an AI Act introduced in April 2021. This legislation categorizes AI systems based on their risk profiles, establishing regulatory measures for high-risk systems and prohibiting those deemed unacceptable. The EU's strategy emphasizes human oversight and data quality, ensuring AI systems operate ethically and reliably. Mandatory assessments facilitate risk identification, aligning AI use with privacy and human rights standards. These efforts are vital for building public trust in AI technologies while mitigating potential abuses of automated decision-making. Also, the European Data Protection Board (EDPB) provides important guidelines and recommendations to candidate countries, helping to implement data protection standards (EDPD, 2024). This is essential to building a cyber security culture that includes not only laws but also good practice.

1.1. Global AI strategies and their impact

Globally, nations are advancing AI technologies with distinct strategies. The United States launched the American AI Initiative in 2019, aiming to maintain its leadership through increased research funding and promoting ethical AI. Meanwhile, China's New Generation AI Development Plan (2017) seeks to position the country as an AI leader by 2030, investing heavily in innovation centers to drive economic growth (Webster et al., 2017).

In Canada, the Pan-Canadian AI Strategy, initiated by CIFAR in 2017, promotes diversity in AI research and supports talent development. This initiative, backed by a $125 million commitment, also delves into the ethical, legal, and social implications of AI (CIFAR, 2017). Japan's Society 5.0 initiative integrates AI with the Internet of Things (IoT), focusing on sustainable development and addressing social challenges. Similarly, Europe's AI efforts are increasingly linked with regulatory and ethical frameworks.

The Stargate project, a $500 billion initiative led by OpenAI and SoftBank, aims to create shared AI computing infrastructure to meet rising demands. The project's first $100 billion phase focuses on facilities in Texas, involving partners like Oracle, Microsoft, and Nvidia. This initiative marks a shift toward shared resources across competitors, supported by the White House, to maintain U.S. AI leadership and address economic and security concerns. However, regulatory challenges regarding market competition may arise (Jackson, 2025). To tackle ethical concerns in AI, organizations should adopt best practices like regular audits, bias detection mechanisms, and a culture of ethical AI development. A clear

accountability framework is essential to address these concerns (Mensah & Sukah Selorm, 2023).

1.2. Regulatory Developments and Compliance in AI

The General Data Protection Regulation (GDPR) underpins EU data protection laws and serves as a model for AI governance in Albania and North Macedonia. Compliance with GDPR ensures AI systems operate ethically, promoting responsible deployment.
The AI Act, passed by the European Parliament, provides a comprehensive framework to regulate AI development, deployment, and use in the EU. It aims to ensure safety, transparency, and the protection of fundamental rights. The Act's risk-based approach categorizes AI systems based on their potential risks, imposing varying obligations depending on their classification. High-risk AI systems face stricter requirements, while general-purpose AI systems are subject to additional regulations to address systemic risks. The Act is expected to come into effect by the end of 2024, with organizations required to map AI systems, conduct risk assessments, and implement governance frameworks to comply with the legislation. In Figure 1 are shown the AI act risk-based approach by EU.

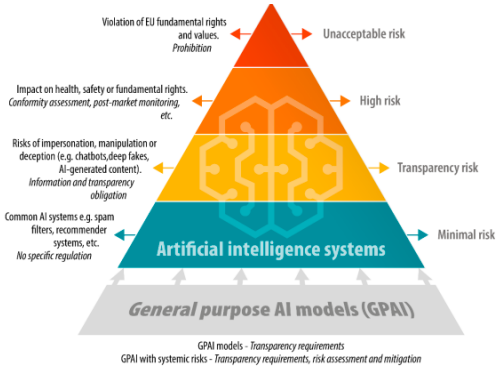


Figure 1. EU AI act risk-based approach (EU, 2024)

2. Current State of AI and Cybersecurity in Albania and North Macedonia

Both Albania and North Macedonia face significant cybersecurity challenges, including increasing cybercrime, outdated infrastructures, and limited enforcement of GDPR. These challenges are exacerbated by the rapid digitalization of government services, escalating threats from state-sponsored cyber activities, and a lack of local expertise in cybersecurity and AI-driven security solutions.

While both countries have made strides toward digital transformation, gaps remain in workforce readiness, regulatory frameworks, and technological capabilities. Key areas requiring urgent attention include:

- Cybercrime Trends: The frequency of cyberattacks targeting government institutions, businesses, and critical infrastructure has risen, revealing weaknesses in cybersecurity policies. Notably, the 2022 cyberattack on Albania's government systems, attributed to state-sponsored actors, disrupted public services and databases (Koloshi, 2022).
- Existing Policies: Both countries have national cybersecurity strategies, but enforcement remains inconsistent. Albania has adopted Laws No. 2/2017 "On Cybersecurity" and No. 9887 "On the Protection of Personal Data," aligning with EU standards. North Macedonia faces similar challenges in policy enforcement.
- Key Priorities: Strengthening regulatory enforcement, investing in AI-driven security measures, and improving cross-border collaboration with EU institutions are essential priorities for both nations.

Addressing these challenges will enhance digital infrastructure protection, foster innovation, and improve trust in AI-driven cybersecurity solutions.

2.1. Development of Artificial Intelligence in Albania and North Macedonia

Both Albania and North Macedonia are advancing AI integration, though challenges remain in aligning with EU standards for data protection and cybersecurity.

Albania has initiated projects to integrate AI into public administration and align with EU legislation. The National Agency for Information Society (AKSHI) is leading an NLP-based project to automate the transposition of EU laws (AKSHI, 2024). Additionally, Albania's National Cybersecurity Agency (NCERT), within the National Agency for Electronic Certification and Cybersecurity (NAECCS), coordinates cybersecurity efforts (NAECCS, 2023). Albania has ratified the Budapest Convention on Cybercrime and enacted laws aligned with international standards to enhance cybersecurity resilience. Notable AI initiatives in Albania include:

- AI for Youth Program: A partnership between the Albanian-American Development Foundation (AADF) and Intel introduces AI education in 30 high schools, aiming to reach 2,000 students by 2026 (AADF, 2021).
- EU Digital Justice Project: This initiative modernizes Albania's justice system through digital transformation, improving efficiency and transparency (En, 2025).
- AI in Public Procurement: The Albanian government has proposed using AI to enhance public procurement transparency, aiming to reduce corruption (Balkaninsight, 2024).
- Memorandum of Understanding with Italy: Albania and Italy have signed a memorandum to strengthen bilateral cooperation in cybersecurity. This agreement focuses on enhancing capabilities to defend against cyber-attacks, sharing information on emerging threats, and adopting best practices in cybersecurity(Caffo, 2024).

Despite these efforts, local businesses in Albania are slow to adopt AI and machine learning, remaining in early stages of AI implementation (Kaso & Xhindi, 2023). To

address this, AKSK has implemented additional training programs for employment, business, and education. Recently, the LAIA project has been introduced to further enhance AI education and align with EU standards (LAIA, 2023).

Similarly, North Macedonia is integrating AI in public administration, digital infrastructure, and services. AI is also increasingly applied in sectors like manufacturing and healthcare. However, challenges remain in raising public awareness and strengthening cybersecurity resilience, especially against geopolitical cyber threats (Poposka, 2023).

A major cybersecurity incident in North Macedonia in 2020 exposed personal data of millions of citizens, highlighting significant gaps in preparedness for large-scale cyber threats. The government's response involved collaboration with law enforcement and private sector cybersecurity firms, along with public transparency efforts. Additionally, both Albania and North Macedonia benefit from IPA (Instrument for Pre-Accession Assistance) II and IPA III funding, which continues to support cybersecurity capacity-building efforts and alignment with EU directives (CILC, 2025).

2.2. Estonia's and Slovenia's Experiences

The integration of emerging technologies such as Artificial Intelligence (AI) with robust cybersecurity frameworks is critical for countries seeking EU membership. As EU members, Estonia and Slovenia provide valuable case studies for Albania and North Macedonia, illustrating how these technologies can be harmonized with national policies to address both digital innovation and cybersecurity challenges.

Estonia stands out for its advanced AI strategy and the 2020 AI Act, which prioritizes AI ethics and data protection. Similarly, Slovenia has aligned its data protection legislation with the GDPR, establishing a strong foundation for both AI development and cybersecurity. The experiences of both countries demonstrate the importance of integrating AI into national policies while simultaneously addressing cybersecurity concerns.

In contrast, Albania and North Macedonia face challenges in modernizing their cybersecurity laws and developing national AI strategies. Albania's cybersecurity framework remains outdated, and North Macedonia's AI development frameworks are still limited. These gaps inhibit their ability to fully capitalize on the potential of AI and secure digital environments for their citizens.

Furthermore, Slovenia's commitment to cybersecurity education—through the establishment of specialized cybersecurity schools and a national coordination center—highlights the importance of building a skilled workforce to address emerging threats. Slovenia is also addressing 5G cybersecurity risks through the implementation of the 5G Cybersecurity Toolbox, further strengthening its digital transformation efforts (Digital Decade Report, 2024).

To overcome these challenges, Albania and North Macedonia should prioritize developing comprehensive AI frameworks that align with EU standards. Strengthening public-private collaboration in cybersecurity will also be essential to enhancing their digital resilience. By drawing lessons from Estonia and Slovenia's approaches, both countries can take significant steps toward securing their digital futures and meeting EU integration criteria.

3. Cybersecurity

Cybersecurity remains a critical concern for both Albania and North Macedonia, especially in light of growing digital threats. Albania, for instance, experienced significant cyberattacks in 2022, notably targeting government institutions. In response, Albania has established key entities such as the National Computer Incident Response Team (AL-CIRT) and the National Authority for Electronic Certification and Cybersecurity (NAECCS) to lead national efforts in mitigating cyber risks. The Albanian National Agency for Information Society (AKSHI) has been taking steps to bolster cybersecurity capabilities, although challenges related to expertise and resources persist. Notably, incidents like the cyberattack on the TIMS (Albanian Immigration System), which compromised sensitive immigration data, and another targeting the e-Albania portal, which disrupted public services, underscored the vulnerabilities in government digital platforms.

In North Macedonia, the National Cybersecurity Strategy (NAECCS) has been instrumental in improving the country's cybersecurity infrastructure and raising awareness about digital risks. The NAECCS closely aligns with EU standards, providing services such as secure online transactions and digital signature verification. Cooperation with both the EU and NATO plays a vital role in enhancing cybersecurity in North Macedonia and Albania.

A comparative analysis of the cybersecurity landscapes in Albania, North Macedonia, Estonia, and Slovenia, as presented in the Global Cybersecurity Index (2024), offers valuable insights into the countries' cybersecurity measures across various domains (Figure 2):

- Legal Measures
- Technical Measures
- Organizational Measures
- Capacity Development

In Figure 2 a), Albania and Estonia show strong legal frameworks in line with international standards, providing a solid foundation for safeguarding data and ensuring cybersecurity. Slovenia also maintains a robust legal structure, reflecting its commitment to effective cybersecurity governance. Albania exhibits relatively strong technical capabilities in cybersecurity, whereas North Macedonia faces considerable challenges in this domain (Figure 2 b), necessitating improvements in technical infrastructure and cybersecurity tools. Both Albania and North Macedonia share comparable organizational measures, reflecting their efforts to establish structured cybersecurity frameworks (Figure 2 c). Estonia’s perfect score for organizational measures exemplifies best practices and serves as a model. However, Albania and North Macedonia lag in capacity development, as highlighted in Figure 2 d), with both countries needing to invest in enhancing their cybersecurity workforce skills. Estonia and Slovenia, in contrast, have made significant investments in workforce development.

Cooperation measures are vital for building resilient cybersecurity frameworks. While Albania has shown notable progress, North Macedonia's lower score underscores the need for stronger international partnerships. Estonia and Slovenia lead by example, demonstrating the importance of collaborative cybersecurity efforts.

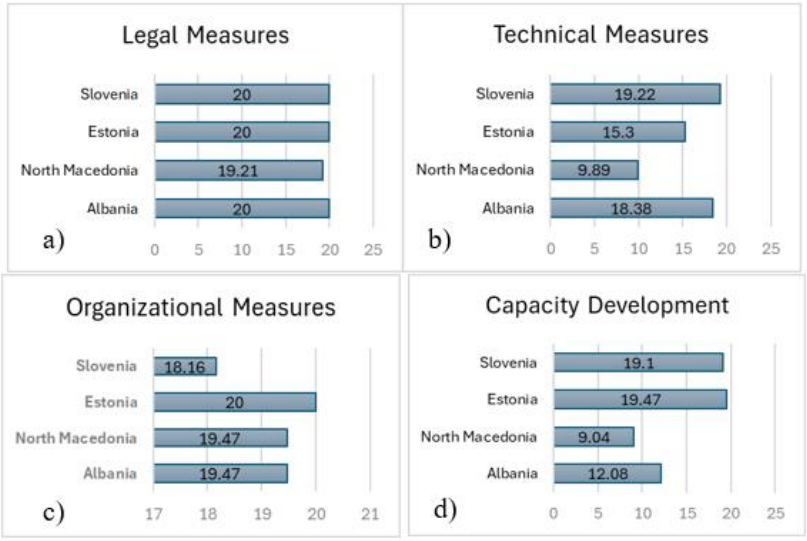


Figure 2. The cybersecurity landscape among Albania, North Macedonia, Estonia, and Slovenia (ITU, 2024)

Estonia and Slovenia, ranked in Tier 1, exhibit the most advanced cybersecurity capabilities, scoring highly across all five pillars: legal, technical, organizational, capacity development, and cooperation. Estonia stands out with technical measures (15.3), organizational measures (20), and cooperation measures (20), while Slovenia achieves similar success in these domains, with technical (19.22), organizational (18.16), and cooperation measures (20).

Albania, in Tier 2, has made significant progress, especially in legal and organizational measures, scoring 20 and 19.47, respectively. However, there is room for improvement in technical capabilities (18.38) and capacity development (12.08), as reflected in its rise of 23 positions in the Global Cybersecurity Index. North Macedonia, placed in Tier 3, faces considerable challenges, especially in technical (8.89) and capacity development (9.04) measures. Though it performs well in legal (19.21) and organizational measures (19.47), further investment and reforms are necessary to raise its cybersecurity standards.

4. Challenges and Recommendations for the Future

Both Albania and North Macedonia face challenges in complying with the General Data Protection Regulation (GDPR). Albania has enacted Law No. 9887 on the Protection of Personal Data, ensuring robust protection measures, but its enforcement is still a work in progress. North Macedonia has similarly introduced data protection regulations, but effective implementation remains a challenge.

Despite their progress, Albania and North Macedonia must continue aligning their regulations with EU standards. Both countries benefit from programs such as Horizon

Europe and Erasmus+, which help improve technological and digital capacities. International cooperation with the EU and NATO is critical for strengthening cybersecurity and data protection frameworks.

Albania is actively advancing its capabilities in artificial intelligence (AI) and cybersecurity. Recent initiatives, such as using Natural Language Processing (NLP) to facilitate EU regulation transposition and deploying a virtual assistant within public administration, underscore Albania's commitment to leveraging AI for governance and regulatory compliance (European Commission, 2023). Moreover, the establishment of AL-CIRT reflects Albania's proactive approach to cybersecurity, aiming to address vulnerabilities and safeguard critical infrastructure. However, challenges such as inadequate data infrastructure and ethical concerns surrounding AI deployment persist.

Both countries must invest in digital infrastructure and professional training to meet EU standards and accelerate their integration process. Strengthening enforcement mechanisms for data protection and cybersecurity regulations is key, as is fostering international collaborations to enhance technological capacities. Currently, AI regulatory frameworks in Southeast Europe are underdeveloped, and harmonization with EU standards is essential to ensure ethical AI use, stakeholder engagement, and adaptive policies that can keep pace with technological advancements (Kovacev et al., 2024).

The OECD highlights the importance of governance principles for AI, focusing on transparency, accountability, and public trust to mitigate AI-related risks (OECD, 2023). Adopting these principles can enhance Albania and North Macedonia's cybersecurity measures and align them with EU standards, addressing the pressing need for effective data protection strategies.

The 2024 AI Index Report suggests several strategies for responsible AI deployment:

- Ethical Governance: Establish ethical frameworks that prioritize human rights and societal well-being.
- Data Privacy Regulations: Enforce robust data protection laws to build public trust in AI systems.
- Public-Private Partnerships: Foster collaborations between government and industry to drive innovation while maintaining accountability.
- Education and Workforce Development: Invest in AI training programs to enhance workforce capabilities and public AI literacy.
- Transparency and Accountability: Require AI systems to be explainable and accountable, with mechanisms for addressing biases.

These strategies will support the responsible adoption of AI, addressing societal challenges and ensuring the technology serves the broader public good (Stanford HAI, 2024).

5. Conclusion

Cybersecurity is a problem in every nation. North Macedonia and Albania Security regulations, knowledge, and procedures are being actively improved by groups like AL-CIRT and NAECCS. Significant strides have been made by both nations to improve their cyber security, particularly in the wake of the assaults. Cyberattacks targeting state institutions and critical infrastructure highlight the urgency of addressing these

vulnerabilities. Achieving EU standards requires continued efforts, international cooperation, and substantial investments in human and technological resources.

The Global Cybersecurity Index 2024 underscores the varied progress in cybersecurity among these countries. While Albania has made considerable progress in legal and organizational measures, it needs to focus on improving its technical capabilities. North Macedonia, while strong in legal and organizational aspects, requires investments in technical infrastructure and enhanced international cooperation. Estonia and Slovenia, ranked in Tier 1, serve as models in cybersecurity, setting benchmarks for Albania and North Macedonia.

Implementing GDPR regulations and adhering to EDPB guidelines will be crucial for both countries as they strive for EU integration. These frameworks will foster a culture of information security and facilitate the adoption of advanced technologies, such as AI, to enhance threat detection and response. A strong commitment to transparency and accountability will be essential for effective data protection and a safer digital environment for citizens. By improving data protection practices and frameworks, Albania and North Macedonia can lead by example and inspire other EU aspirants.

Furthermore, the European community is increasingly emphasizing the advancement of the Western Balkans. The institutions set up in both Albania and North Macedonia are growing their influence and impact. Efforts in training, legal framework implementation, investments, and raising awareness among citizens, businesses, and employees are improving. However, some ongoing challenges include insufficient human resources and outdated legal frameworks, and the focus is on improving these areas to meet EU standards. Strengthening regional cooperation and learning from one another will be essential for fostering a collaborative cybersecurity environment in the Western Balkans. By adopting a proactive and unified approach, Albania and North Macedonia can improve their cybersecurity posture and continue their path toward EU membership.

References

AADF (Albanian American Development Foundation) (2021). Memorandum of understanding paves way for AI empowerment among Albanian youth. Albanian American Development Foundation. https://www.aadf.org/memorandum-of-understanding-paves-way-for-ai-empowerment-among-albanian-youth

Balkan Insight (2024). Using AI in Albanian public procurements: No easy solution for corruption. https://balkaninsight.com/2024/10/28/using-ai-in-albanian-public-procurements-no-easy-solution-for-corruption

Caffo, A. (2024). Italy and Albania sign a memorandum for cooperation in cybersecurity. 4iMAG. https://4imag.com/italy-and-albania-sign-a-memorandum-for-cooperation-in-cybersecurity

Centre for International Legal Cooperation (CILC). EU support to Western Balkans cybersecurity capacity building. CILC. https://www.cilc.nl/projects/eu-support-to-western-balkans-cybersecurity-capacity-building/

EDPD-European Data Protection Board (2024). European Data Protection Board (EDPB). https://www.edpb.europa.eu/edpb_en

European Commission (2021). *Artificial Intelligence Act*.

European Commission (2023a). *Albania 2023 report: Accompanying the document communication from the commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions: 2023 communication on EU enlargement policy (SWD(2023) 690 final).* Brussels.

European Commission (2023b). *EU AI Act: First regulation on artificial intelligence.* https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence

European Commission (2024a). *European approach to artificial intelligence.* https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence

European Commission (2024b). *European AI office*. https://digital-strategy.ec.europa.eu/en/policies/ai-office

European Parliament (2023). *Cybersecurity strategy*.

European Union (2022). General Data Protection Regulation (GDPR). https://eur-lex.europa.eu/EN/legal-content/summary/general-data-protection-regulation-gdpr.html

Gartner (2024). *AI readiness.* https://www.gartner.com/en/information-technology/topics/ai-readiness

GDPR Info (2021). *General data protection regulation compliance guidelines.*

International Telecommunication Union (ITU) (2024). Global cybersecurity index 2024 (5th ed.). https://doi.org/10.1787/2476b1a4-en

Jackson, A. (2025). How the $500bn Stargate venture signals an AI strategy shift. Data Centre Magazine. https://datacentremagazine.com/technology-and-ai/how-500bn-stargate-venture-signals-ai-strategy-shift

Koloshi, E. (2022). Session I: Conducting criminal investigations of ransomware attacks. *International workshop on conducting criminal investigations of ransomware attacks*, The Hague, Netherlands. Council of Europe. https://rm.coe.int/session-i-edmond-koloshi-albania/1680a8cbe4

Kaso, E., Xhindi, T. (2023). The use of artificial intelligence and machine learning technology by companies in Albania. UMSH Press. https://www.umsh.edu.al/media/650d6a5695eea.pdf

Kovacev, A., Vujanović, P., Stanković, M. (2024). Regulatory frameworks for AI in Southeast Europe: Current state and future directions. *Journal of Regulation & Governance*, **22**(1), 22-34. https://www.eca.europa.eu/en/publications/SR-2024-08

LAIA (2023). *LAIA project: Developing AI education in Albania and Kosovo.* ERASMUS-EDU-2023-CBHE. https://laiaproject.eu/

Mensah, G. B., Sukah Selorm, J. M. (2023). Addressing ethical concerns in artificial intelligence: Tackling bias, promoting transparency and ensuring accountability. https://doi.org/10.13140/RG.2.2.20173.61925

National Agency for Electronic Certification and Cyber Security (NAECCS). (2023). *Annual report 2023*. https://aksk.gov.al/en/annual-report-2023/

National Agency for Information Society (AKSHI) (2024). *Report on the use of AI for transposing EU legislation*.

National Authority for Electronic Certification and Cyber Security (NAECCS) (2024). Law No. 25/2024 on Cybersecurity (in Albanian). https://aksk.gov.al/wp-content/uploads/2024/04/ligj-2024-03-21-25-5.pdf

OECD (2023). *AI governance: Implementing the OECD principles on artificial intelligence. OECD Artificial Intelligence Papers, No. 22.* https://doi.org/10.1787/2476b1a4-en

Poposka, V. (2023). Normative framework toward cyber crimes in North Macedonia. *International Scientific Journal Sui Generis*, **2.** 85-99. 10.55843/SG2321085p.

Rich, E., Knight, K., Nair, S. B. (2009). *Artificial intelligence (3rd ed.)*. Tata McGraw-Hill.

Russell, S., Norvig, P. (2010). *Artificial intelligence: A modern approach.* Prentice Hall.

Stanford University (2024). *2024 AI index report.* Stanford Human-Centered AI Institute. https://aiindex.stanford.edu/wp-content/uploads/2024/04/HAI_2024_AI-Index-Report.pdf

UK Government (2024). *Cyber security breaches survey 2024*. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2024/cyber-security-breaches-survey-2024

Webster, G., Creemers, R., Kania, E., Triolo, P. (2017). *Full translation: China's 'New generation artificial intelligence development plan' (2017).* Stanford University. https://digichina.stanford.edu/work/full-translation-chinas-new-generation-artificial-intelligence-development-plan-2017/

Received October 24, 2024, revised February 19, 2025, accepted February 19, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 177–199
https://doi.org/10.22364/bjmc.2025.13.1.10

Performance-Driven and Cost-Efficient Convergence of Cloud and HPC: Evaluating MinIO and LustreFS

Hrachya ASTSATRYAN[1], Hovhannes BAGHDASARYAN[1,2], Ruben ABAGYAN[3], Hovakim GRABSKI[3,4], Siranuysh GRABSKA[3,4]

[1] Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
[2] Department of Informatics and Computer Engineering, International Scientific-Educational Center of NAS RA, Yerevan, Armenia
[3] Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, San Diego, CA, USA
[4] L.A. Orbeli Institute of Physiology of NAS RA, Yerevan, Armenia

baghdasaryan@ieee.org, abagyan@health.ucsd.edu, hgrabski@health.ucsd.edu, sgrabska@health.ucsd.edu, hrach@sci.am

ORCID 0000-0001-8872-6620, ORCID 0009-0008-3242-5464, ORCID 0000-0001-9309-2976, ORCID 0000-0001-6115-9339, ORCID 0000-0001-9291-3357

Abstract. Integrating cloud technologies into high-performance computing (HPC) systems addresses the rapidly increasing demand for data growth in HPC. The HPC over cloud solutions may simplify complicated workflow, enhance scalability, and improve the end-user experience. Reliable file systems in HPC over cloud environments may efficiently manage vast amounts of data. This study evaluates various approaches to distributed storage and cloud object storage performance. A cost-effectiveness and performance analysis prove that cloud-integrated HPC can offer a scalable alternative to traditional storage solutions.

Keywords: cloud, distributed file system, object storage, Lustre, S3 API

1 Introduction and background

Parallel file systems are fundamental components of high-performance computational (HPC) infrastructure, providing the necessary storage, organization, sharing, protection, and performance capabilities to support the complex computational workflows and data-intensive applications prevalent in the HPC domain (Lockwood et al., 2018). The parallel file systems effectiveness in HPC encompasses scalability, cost-effectiveness, reliability, performance, and manageability. The end of Dennard scaling and Moore's

Law has made it challenging to scale HPC systems within a given performance range, especially in large systems such as supercomputers (Milojicic et al., 2021). Many vendors have deployed scalable cloud object stores to accommodate the continued growth of unstructured data and simplify access to HPC.

Object stores offer faster resource access and cost reduction than traditional file systems. The object stores scale performance and capacity efficiently without hierarchical structures, ensuring seamless growth with demand. Object stores' "key-value" data format enables robust data protection through replication and erasure coding, enhancing durability and reliability. Object stores like Amazon S3 and Google Cloud Storage are widely used for their efficient resource access and scalability without hierarchical structures (Palankar et al., 2008). Leveraging a "key-value" data format, the stores ensure data protection through replication and erasure coding, bolstering durability and reliability. Amazon S3 is a highly reliable and widely used object storage service that offers seamless scalability and comprehensive data protection features. Numerous cloud storage providers offer a RESTful gateway broadly compatible with the S3 interface.

Many libraries face limitations in harnessing the full potential of object storage due to their dependence on traditional file system interfaces like POSIX (Portable Operating System Interface). The POSIX programming API defines a set of operations for interacting with files, directories, and entire file systems. This integration challenge often results in storage sprawl, as object stores are frequently deployed alongside file systems, resulting in ad hoc data access and management across both systems. Storage sprawl causes numerous problems, including over-provisioning, reduced backup efficiency, and cost inflation (Smith, 2016). Managing data complexity across multiple storage systems increases administrative overhead and operational costs. Additionally, the need for redundant storage resources and backup infrastructure further escalates operational expenses. In addition, the coexistence of object storage and traditional file systems often leads to redundant provisioning of storage resources to accommodate both systems' requirements, resulting in wasted resources and increased costs.

The convergence of cloud and HPC storage architectures can effectively address modern storage system challenges to seamlessly integrate object and file storage, consolidating data onto a unified platform (Lofstead et al., 2016). It seeks to enhance storage capabilities while preserving established semantics and interfaces, ensuring consistent and efficient data access across diverse computing environments. Various approaches exist to realize this convergence. One involves integrating S3-compatible gateways with parallel file systems at the storage level (Gadban and Kunkel, 2021). Another approach extends distributed file systems by combining them with cloud object storage (Luettgau et al., 2023). Additionally, efforts enhance the performance of existing cloud-native distributed file systems (Jeong et al., 2019). These strategies bridge the gap between traditional file systems and object storage in converged cloud and HPC environments.

The article studies several cloud and HPC systems' converging concepts and efficient solutions, focusing on cost-effectiveness and performance metrics. Specifically, we evaluate MinIO and LustreFS (Schwan et al., 2003) against S3 object storage to ascertain their suitability in achieving this convergence. The structure of this paper is as follows. Section 2 presents state-of-the-art and related work. The methodology and

experimental environment are described in Section 3. Section 4 contains a performance analysis results. Finally, the last Section 6 provides the conclusion of our study.

2 Related work

In recent years, numerous publications have delved into the convergence of cloud object storage and high-performance file systems, including those by Chen, Li and Ke (2017), Lackschewitz et al. (2022), Huang et al. (2015), Durner et al. (2023), and Liu et al. (2018). These studies shed light on the strengths and limitations of various approaches. Jones et al. (2017) evaluated the high-performance parallel file system Lustre and Amazon's S3, highlighting S3's suitability for sharing vast amounts of data over the Internet. The results show that with proper implementation of parallel I/O, full network bandwidth performance can be attained, ranging from 10 gigabits/s over a 10 GigE S3 connection to 0.35 terabits/s using Lustre on a 1200 port 10 GigE switch. Lustre excels in processing large datasets locally. However, their study did not assess the performance of a file system compatible with the S3 API for HPC environments. In Gadban et al. (2020) study, they examined the RESTful API protocol via HTTP for high-performance file storage, contrasting it with the HPC-native communication protocol Message Passing Interface (MPI) in object storage operations. The authors showed that REST often delivers comparable latency and throughput to MPI implementations. Still, their study did not assess its suitability, efficiency, and performance in handling large file accesses. Notably, their research did not explicitly explore the performance implications of integrating a cloud storage system like S3 within an HPC environment.

Several studies propose implementations aimed at achieving such convergence, such as MarFS (Inman et al., 2017; Chen, Grider and Montoya, 2017) and ArkFS (Cho et al., 2023) scalable distributed file systems designed to be near-POSIX compliant, operating on top of object storage systems. They support S3-compatible platforms with the aid of suitable API translation modules. However, they do not fully adhere to POSIX standards. Many lack support for essential POSIX features like symlinks, hardlinks, or file attributes (chmod), leading to suboptimal performance for random writes (Lillaney et al., 2019).

According to the market survey, supporting POSIX completeness is extremely important for HPC, as most datacenters need to support legacy applications that rely on POSIX semantics (Inman et al., 2017). Therefore, there is the most significant interest in convergence with POSIX-complete file systems. This study addresses the limitation by examining well-established parallel file systems and object storage solutions supporting the S3 API. Among the parallel file systems commonly employed in HPC environments, three widely recognized Free and open-source software systems — Lustre, BeeGFS (formerly known as FhGFS), and DAOS (Distributed Asynchronous Object Storage) were studied. These systems frequently appear in submissions to the IO500 ranking, a semi-annual performance evaluation of HPC storage systems. IO500 evaluates the storage system performance based on bandwidth and metadata performance.

The studies (Lackschewitz et al., 2022; Manubens et al., 2022; Hennecke, 2020) show that DAOS outperforms other parallel file systems in most metrics. At the time of

Table 1. The number of DAOS and Lustre file systems in ISC24 List

ISC24 List	Lustre	DAOS
TOP-25	1	9
TOP-50	7	16
TOP-100	23	20

writing, DAOS-based systems occupy nine positions in the top 25 in IO500 Research ISC24 list[5]. Table 1 shows the number of installations with DAOS and Lustre.

DAOS requires large NVMe (non-volatile memory express) and NVRAM (non-volatile random access memory) devices, making it unsuitable for all environments. In the near future, the relatively high prices of these storage devices will limit the use of DAOS in datacenters and especially in research. With this in mind, the most promising avenues lie in the convergence of cloud storage systems with Lustre (Gadban, 2022). It is worth mentioning that Ceph offers greater redundancy than Lustre, but Lustre is faster in an HPC environment. However, Lustre can be problematic due to its single point of failure.

Table 2. Storage Solution Comparison Matrix

	LustreFS	BeeGFS	JuiceFS	Ceph	MinIO	DAOS
File storage support	yes	yes	yes	yes	no	no
Block storage support	no	no	no	yes	no	no
S3 support	no	no	yes	yes	yes	no
POSIX compliance	high	high	high	acceptable	low	relaxed
Scalability	high	high	low	medium	low	high
High availability	yes	yes	yes	yes	yes	yes
License	GPLv2	GPLv2	Apache License 2.0	LGPLv3	AGPLv3	BSD-2

Comparative characteristics based on preliminary research and available documentation for all considered file systems are presented in Table 2.

[5] IO500 ISC24 Research List - `https://io500.org/list/isc24/io500`

3 Methodology

The methodology section provides an overview of the parallel distributed file systems under examination, describes the benchmarking tools employed, and explains the approach used for cost-effectiveness analysis.

3.1 Overview of parallel distributed file systems

A selection was made of Lustre in combination with the MinIO object store to explore further the convergence approach of integrating S3-compatible gateways with parallel file systems at the storage level. The lightweight architecture of MinIO and its full S3 compatibility make it an optimal candidate for deployment in gateway mode or as a complement to Lustre in hybrid storage environments. As an alternative approach, JuiceFS was considered due to its combination of POSIX compliance and native S3 compatibility. Additionally, Ceph was evaluated as a highly versatile storage system that is effectively used for data redundancy and fault tolerance.

This section presents the overview of Ceph, Lustre, MinIO, and JuiceFS parallel distributed file systems.

3.1.1 Ceph Ceph is a distributed object storage and file system with excellent performance, reliability, and scalability (Weil, Brandt, Miller, Long and Maltzahn, 2006).

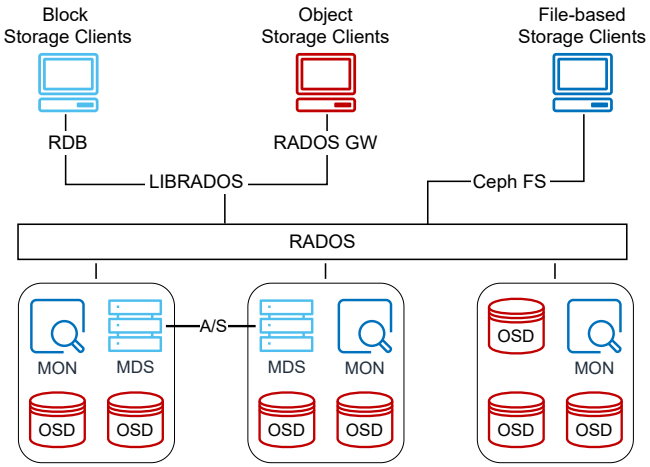


Fig. 1. Ceph architecture.

It operates on a cluster of commodity hardware, utilizing a scalable architecture for seamless expansion as storage requirements grow. Ceph organizes data into objects stored within logical pools. Each is managed independently to optimize performance and reliability. In a Ceph cluster, nodes fulfill three different roles:

- **MDS (Metadata Server)** - act as a metadata service required for the Ceph.
- **OSD (Object Storage Daemon)** - serves as storage resource provider responding to client requests and ensuring data synchronization with other OSD nodes.
- **MON (Monitor)** - responsible for monitoring the overall status of the entire Ceph cluster.

One of the critical components of Ceph is its RADOS (Reliable Autonomic Distributed Object Store) architecture, which ensures data redundancy and fault tolerance by replicating objects across multiple nodes in the cluster (Van der Ster and Wiebalck, 2014). This redundancy enhances data durability and enables high availability and resilience to node failures. In addition to its object storage capabilities, Ceph provides a POSIX-compliant distributed file system called CephFS. CephFS allows users to mount Ceph storage as a traditional file system, enabling seamless integration with existing applications and workflows that rely on standard file access protocols. The LibRados programming interface serves as a basic framework for various client interfaces.

As shown in Fig. 1, Ceph's object storage is exposed through the RADOS gateway, the block storage through the rados block device, and the file system is exposed through Ceph FS. All of these components rely on Librados interfaces for their operation. Ultimately, the data is stored as objects in the RADOS system. Ceph uses the CRUSH (i.e., Controlled Replication Under Scalable Hashing) algorithm (Weil, Brandt, Miller and Maltzahn, 2006) to ensure that data is distributed evenly across the cluster, allowing for easy retrieval by all cluster nodes.

Overall, Ceph could be considered an ideal cloud solution for HPC if there was no performance degradation when scaling the system (Gudu et al., 2014). There are very few studies (Jeong et al., 2019; Zhang et al., 2019; Li et al., 2020) on improving Ceph performance, although this could be a promising direction in the convergence of cloud and HPC.

3.1.2 Lustre Lustre has been one of supercomputers' most popular parallel distributed file systems for many years, offering scalability, high throughput, and low latency. Lustre's architecture was carefully designed to serve as a scalable storage platform for computer networks, using a distributed, object-based storage approach. A Lustre consists of three key components:

- **Metadata Servers (MDS)** - host metadata targets (MDT) per Lustre file system and manage namespace metadata such as file names, access permissions, etc.
- **Object Storage Servers (OSS**) - store file types on object storage target (OST) devices. A Lustre file system is the total capacity of its OSTs.
- **Clients** - access and use the data. Lustre offers a unified namespace for all files and enables standard POSIX semantics.

The typical architecture of Lustre is shown in Fig. 2.

In most scenarios, the MDT, OST, and client components are distributed across different nodes within a Lustre file system and connected over a network. The Lustre Network (LNet) offers compatibility with various network connection options, including InfiniBand connections, Omni-Path, or TCP/IP over Ethernet.

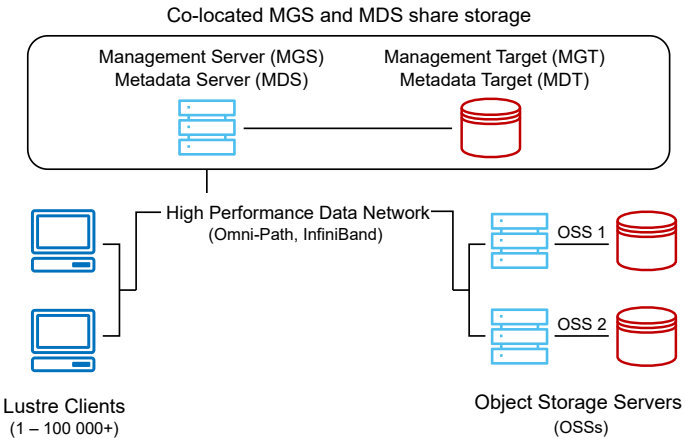


Fig. 2. Lustre architecture.

Lustre provides a coherent, global POSIX-compliant namespace for large-scale computer infrastructure, including the world's largest supercomputer platforms. It can support hundreds of petabytes of data storage and tens of terabytes per second in simultaneous, aggregate throughput (Panda et al., 2022). Such convergence would be promising if the S3 gateway adapts to HPC (Gadban and Kunkel, 2021).

3.1.3 MinIO and JuiceFS MinIO is an object storage solution compatible with the Amazon Web Services S3 API and encompasses the full range of S3 core functionality. MinIO achieves horizontal scalability using a concept known as Server Pools. Server pools integrate technology components, each representing a self-contained group of nodes with computing, networking, and storage resources. In addition, MinIO provides extensive functionality for working with metadata, which is valuable from the end user's perspective (Spiga et al., 2022). Given the S3 compliance, running MinIO in the gateway mode in front of Lustre is one of the convergence scenarios.

Another approach is to implement high-performance POSIX-compliant distributed file systems. One such solution is JuiceFS, an open-source distributed file system compatible with POSIX, Hadoop distributed file system (HDFS) and S3 protocols. Full POSIX compatibility allows almost all kinds of object storage, such as Ceph or MinIO. JuiceFS offers data management, analysis, archiving, and backup APIs (Luettgau et al., 2023). Therefore, JuiceFS integration with MinIO is interesting for further exploration.

JuiceFS is compatible with POSIX, HDFS, and S3 protocols, making it highly versatile and suitable for various use cases. The JuiceFS architecture (Fig. 3) consists of three key components: the metadata engine, the object storage backend, and the client layer. The metadata engine implemented with Redis manages file system metadata such as file hierarchy, permissions, and attributes. The speed and efficiency of Redis make it the ideal choice for this purpose, ensuring that metadata operations do not remain a bottleneck even in large-scale deployments. The object storage backend is where JuiceFS

stores data blocks. JuiceFS, supporting various backends, is flexible in choosing suitable storage solutions. The client layer provides access to the file system through a POSIX-compliant interface. JuiceFS also supports additional protocols, such as NFS and SMB, further improving compatibility.

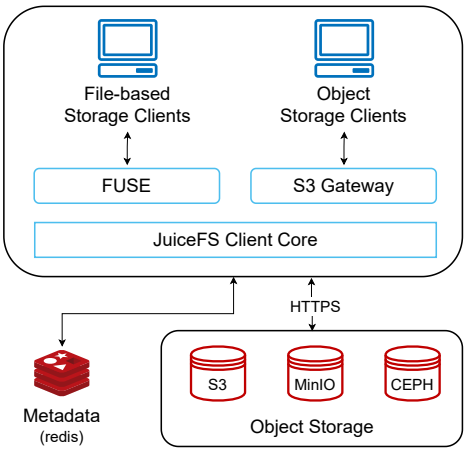


Fig. 3. JuiceFS Architecture.

The integration of JuiceFS and MinIO creates a robust storage solution that combines the scalability and cost-effectiveness of object storage of MinIO with the ease of use and performance of JuiceFS. JuiceFS is an interface layer in this architecture that connects applications to the underlying MinIO storage. The Redis manages the metadata, while MinIO stores and splits data into blocks. The integration is beneficial for optimizing performance through client-side caching. JuiceFS clients cache frequently accessed files locally, significantly reducing latency and increasing throughput for read-intensive workloads. This feature is particularly beneficial in machine learning applications, where large data sets are frequently accessed repeatedly (Luettgau et al., 2023). MinIO's horizontal scalability also allows for handling increasing amounts of data without sacrificing performance. This scalability is achieved by dynamically adding nodes to server pools.

JuiceFS's full POSIX compliance makes it an excellent choice for organizations moving from legacy systems to modern, cloud-native storage. POSIX compliance ensures that applications requiring a traditional hierarchical file system can run seamlessly on JuiceFS without significant changes. This compatibility also extends to MinIO, as JuiceFS translates traditional file operations into object storage operations, enabling efficient use of MinIO as a backend. This design allows organizations to leverage the cost benefits of object storage while maintaining the ease of use of traditional file systems. Additionally, the modular nature of this architecture allows the metadata and data storage components to scale independently, enabling optimal resource utilization.

3.2 Benchmarking tools

COSBench (Cloud Object Storage Benchmark) and MDtest are used to evaluate the performance of storage systems. Each tool offers unique capabilities and focuses on different aspects of storage performance assessment (Zheng et al., 2012).

3.2.1 COSBench COSBench tool developed by Intel evaluates the performance of cloud object storage services, such as Amazon S3, OpenStack Swift, and Ceph RADOS Gateway. The tool simulates workloads and measures key performance metrics, including throughput, bandwidth, latency, and scalability. COSBench evaluates cloud object storage's read and write performance as a system compatible with the S3 protocol. The COSBench interface includes three core operations (create, get, and delete an object) for identifying bottlenecks and measuring capacity.

3.2.2 MDtest (IOR) The IOR, developed by Lawrence Livermore National Laboratory, generates various I/O patterns to evaluate the throughput and latency of storage systems, such as sequential and random reads/writes. IOR supports configurable parameters such as block or file sizes or several processes to tailor the benchmark to specific use cases. It provides detailed performance metrics, including bandwidth, IOPS (I/O operations per second), and access latencies. MDtest is part of the IOR suite that evaluates metadata performance in parallel file systems. It focuses on measuring the performance of metadata operations such as file creation, deletion, and listing. MDtest allows users to generate metadata workloads, including small and large file counts, directory hierarchies, and metadata access patterns. It provides detailed metrics on metadata throughput, latency, and scalability.

3.3 Experimental environment

All experiments were conducted utilizing the CloudLab (Duplyakin et al., 2019), a collaborative initiative involving five US universities and US Ignite, offering robust testbeds tailored for the computer science research community.

Servers within the University of Wisconsin cluster with the c220g2 configuration were selected for the experiments. The configuration details are provided in Table 3.

Table 3. Node configuration of c220g2

CPU	Two Intel E5-2660 v3 10-core CPUs at 2.60 GHz (Haswell EP)
RAM	160 GB ECC Memory (10x 16GB DDR4 2133 MHz dual rank RDIMMs)
Disk 1	One Intel DC S3500 480 GB 6G SATA SSD
Disk 2	Two 1.2 TB 10K RPM 6G SAS SFF HDDs
NIC	Dual-port Intel X520 10GB NIC (PCIe v3.0, 8 lanes)
NIC	Onboard Intel i350 1GB

3.4 Cost evaluation

In this study, cost predictions were made using the AWS Pricing Calculator to estimate expenses associated with various AWS services. The calculations are based on the widely adopted Pay-as-you-go (PAYG) model, wherein users are billed monthly for the specific services they utilize. This approach ensures transparency and accuracy in forecasting expenses, allowing researchers to plan and manage their budgets effectively.

The total cost model can be represented by equation (1), where:
I – total number of connected services.
N – total number of consumed resources from the service i.
p_{ni} – unit price for consumed resource n from the service i.
q_{ni} – quantity of units for consumed resource n from the service i.

$$C_{\text{cloud}} = \sum_{i=1}^{I} \left(\sum_{n=1}^{N} p_{ni} \cdot q_{ni} \right). \tag{1}$$

Such a model is beneficial for small or short-term projects. However, the costs can be excessive for large HPC research projects dealing with large volumes of data, even if the vendor offers significant discounts and cost optimization tools. Therefore, most research studies (Smith et al., 2019; Emeras et al., 2016) on this topic concluded that running scientific workloads on-premises is more cost-effective than running them in the cloud. The main cost factors of on-premises implementation of HPC are shown in Table 4.

Table 4. Main Cost Factors

Capital expenditures (CapEx)	Operating expenditures (OpEx)
Servers	Electricity
Network	Staff
Storage	Maintenance
Facilities	Depreciation
Licenses	Recurring licenses

The equation (2) for total spending over M months can be expressed as follows:

$$C_{\text{on-premises}} = \sum_{m=1}^{M} \left(\frac{C_{CapEx}}{m} + C_{OpEx} \right). \tag{2}$$

The calculations in (Gadban, 2022) show that for $M \geqslant 12$, using an on-premise solution is more advantageous than using the cloud. Furthermore, this inequality becomes even more significant in regions with lower operating costs (including space rental, salaries, and wages).

4 Performance analysis

This section describes the configuration of benchmarking tools and the experimental environment for studying the throughput and average response time for the main operations with the file systems. It presents the results of the experiments carried out.

4.1 Configuration of benchmarking tools and environment

4.1.1 COSBench configuration The COSBench testing procedure uses a workload configuration file to simulate different usage patterns. A workload is represented as an XML file specifying important testing parameters. Key components of this configuration include Special Work blocks and Operation blocks:

Special Work blocks:

```
<workstage name="<name>">
  <work
    type="init|prepare|cleanup|dispose"
    workers="<num>"
    config="<key>=<value>;..." />
</workstage>
```

Operation blocks:

```
<work name="<name>"
  workers="<num>"
  runtime="<num>">
    <operation type="read|write|delete"
    ratio="<1-100>"
    config="<key>=<value>;..." />
</work>
```

Among the parameters that define the configuration, the following should be highlighted:

- **workers**: number of threads to conduct the work in parallel
- **runtime**: duration of the work
- **sizes**: object size with unit (B/KB/MB/GB)

Over 120 configuration files were created for the test environment to simulate various usage patterns, comprehensively analyzing system performance under different conditions.

COSBench evaluates performance using several crucial metrics: average response time (Avg-ResTime), operation count (Op-Count), and throughput. Avg-ResTime measures the average time to complete an operation from request initiation to response. This metric indicates system latency, with lower values signifying higher responsiveness. It helps identify and reduce performance bottlenecks. Op-Count is the total number of operations (read, write, delete) executed during testing. Throughput is the amount of data processed per unit of time, typically measured in bytes per second (B/s). This metric

gauges data transfer efficiency and helps determine optimal configurations for maximizing data processing speed. Analyzing Avg-ResTime, Op-Count, and Throughput helps evaluate the system's performance.

4.1.2 MDTest configuration The first step is to clone the repository and build the application by executing the following commands:

```
$ git clone https://github.com/hpc/ior.git
$ cd ior
$ ./bootstrap
$ ./configure
$ make
```

Then to run MDtest, it is performed the following command:

```
mpirun -n <...> mdtest -n 10000000 -w 3901 -e 3901
```

The main parameters for MDTest configuration include:

- **mpirun -n**: initiates MPI processes
- **-n 1000000**: each process will create 1 000 000 files and directories
- **-w 3901**: writes 3901 bytes to each file after its creation
- **-e 3901**: reads 3901 bytes from each file

These configuration settings evaluate the system's performance under different conditions.

4.1.3 Environment configuration Clusters were deployed using disk images based on Linux kernel version 4.18.x for the experiments, widely adopted by Rocky Linux 8.x and CentOS distributions. This approach ensures compatibility with existing software and hardware configurations. The Ceph setup included an MDS, a MON, and various OSDs, while the Lustre cluster consisted of an MDS and a number of OSSes. The JuiceFS cluster was configured, ensuring that the setup mirrored Ceph and Lustre for evolution (Dai et al., 2019).

The selected configurations are critical for understanding the performance characteristics by considering key performance metrics and behaviors to evaluate various loads and operational scenarios. The evaluation results highlight the capabilities and limitations of each storage solution, guiding future improvements and optimizations.

4.2 Results

4.2.1 COSBench The results are shown in Fig. 4. The highest throughput for Lustre+MinIO is 488.1 – 510.32 ops/s. In addition, Lustre+MinIO has the best average response time performance. Notably, the average response time changes insignificantly as the workload increases, indicating the high scalability of this solution.

Anomalous results were observed for block sizes of 4KB and 4MB, likely due to the minimum chunk size requirement of 5MB for S3. A correlation was noted between

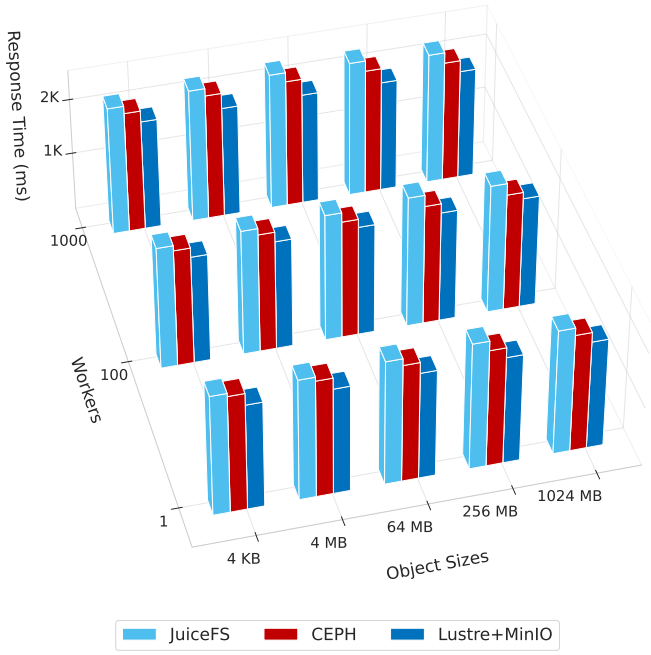


Fig. 4. Average Response Time for the write operation

the variables under consideration for larger chunk sizes. In the case of Ceph, as block sizes increase, the average response time tends to rise, indicating longer durations for write operations. Conversely, throughput increases with larger block sizes and a more significant number of workers. This suggests enhanced efficiency under conditions of higher parallelism and more extensive data sizes.

Similarly, JuiceFS exhibits comparable trends, with the average response time increasing as block sizes and the number of workers rise. However, JuiceFS generally shows a longer average response time than Ceph under similar configurations, highlighting potential performance differences between the two storage systems. In the Lustre with MinIO setup, average response time, and throughput vary with block sizes and worker counts. This configuration demonstrates competitive performance metrics, particularly notable for their lower average response time and higher throughput in specific scenarios compared to Ceph and JuiceFS (see Fig. 5).

4.2.2 MDTest During test iteration, each MPI task generates, parses, and deletes a specified number of directories and files while measuring operation performance per second (ops/s).

The test configuration was designed following the IO500 benchmark guidelines, specifically utilizing variations of mdtest-hard.

The main statistical performance metrics for storage operations are presented in Table 5.

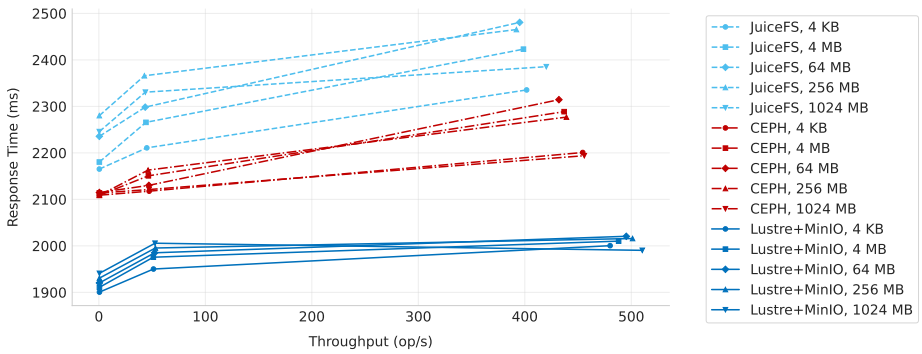


Fig. 5. Throughput and average response time.

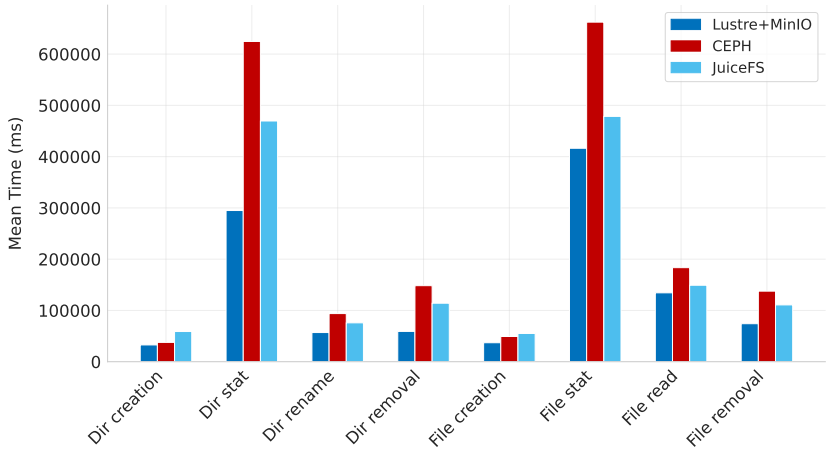


Fig. 6. Mean Performance Comparison of MDTest by Operations

MDTest allows assessment of the maximum, minimum, and average time values for operations such as creation, statistics, rename, and removal for both files and directories. For clarity, Fig. 6 shows averaged metrics across all test iterations on each node for each operation.

As expected, Lustre significantly outperforms its counterparts across all operations.

5 Cost analysis

5.1 Cloud costs

This section develops a cost model to evaluate the costs associated with using Amazon AWS for data processing and cloud storage. The proposed model considers core AWS services, which include Amazon FSx for Lustre, Amazon S3, and Elastic Load Balancing (ELB). These services provide scalable solutions for file storage, object storage,

Table 5. Performance metrics for storage operations, mean time (ms)

Operation	Storage	Max	Min	Mean	Median
Directory creation	Lustre	32744.976	14104.531	23109.016	24385.488
	Ceph	37694.494	28103.387	30661.326	28103.387
	JuiceFS	58899.664	36306.540	44500.488	38396.262
Directory stat	Lustre	294895.873	2815347.026	1471021.501	2815347.026
	Ceph	624617.126	5208374.519	2004635.763	5208374.519
	JuiceFS	469476.606	4139251.949	1543666.445	4139251.949
Directory rename	Lustre	56912.046	1916.461	29414.924	1916.461
	Ceph	93897.423	19557.603	37198.115	19557.603
	JuiceFS	75820.315	43232.581	59567.710	43232.581
Directory removal	Lustre	59122.169	6621.153	32821.761	6621.153
	Ceph	148323.927	98359.250	83700.888	98359.250
	JuiceFS	114115.195	70459.051	91608.189	70459.051
File creation	Lustre	72868.254	1055.402	39469.875	1055.402
	Ceph	49197.739	89360.672	40968.084	89360.672
	JuiceFS	55114.241	79455.935	66175.250	79455.935
File stat	Lustre	4422505.272	6423818.786	2187457.716	6423818.786
	Ceph	662293.384	3094057.244	1222599.920	3094057.244
	JuiceFS	478474.104	4103614.128	1591060.817	4103614.128
File read	Lustre	134355.308	9864.991	62187.950	9864.991
	Ceph	183445.766	897273.291	290572.401	897273.291
	JuiceFS	149146.718	1084472.024	451627.622	1084472.024
File removal	Lustre	125090.710	5420.850	63696.992	5420.850
	Ceph	137649.043	177107.122	115788.872	177107.122
	JuiceFS	110802.135	137756.189	124121.737	137756.189

and load balancing. AWS follows a PAYG pricing model, allowing users to pay only for the consumed resources, with no upfront costs. Discounts of up to 50% are available through reserved or spot instances; however, these options are not considered in this model due to the general unpredictability of resource requirements in most workflows.

The resources included in the model are evaluated based on key pricing components: storage capacity, throughput capacity, backup storage, data transfer, and processing costs. Specific pricing details for each service, excluding compute nodes, which would also be deployed in the cloud environment, as outlined in Table 6.

Table 6. AWS Services Pricing Overview

AWS Service	Pricing Details (USD/GB/month)
FSx for Lustre	
HDD Storage	0.088
SSD Storage	0.794
Backup Storage	0.054
Elastic Load Balancing	
Application Load Balancer	0.008
Network Load Balancer	0.006
Data Processing	0.008
Amazon S3	
Standard Storage (First 50 TB)	0.0245
Standard Storage (Over 50 TB)	0.0235
Glacier Deep Archive	0.00099

Each service contributes specific cost elements based on usage metrics, consolidated into an overall cost model for a comprehensive assessment.

Amazon FSx for Lustre, a service designed for high-performance workloads, incurs costs for HDD and SSD storage, throughput capacity, and backup storage. These costs are monthly TB for storage and megabytes per second per tebibyte (MBps/TiB) for throughput. Therefore, the cost of Amazon FSx for Lustre, costs includes storage, throughput, and backup, expressed as:

$$C_{\text{FSx}}(m) = \left(p_{\text{HDD}} \cdot q_{\text{HDD}} + p_{\text{SSD}} \cdot q_{\text{SSD}} + p_{\text{Throughput}} \cdot q_{\text{Throughput}} + p_{\text{Backup}} \cdot q_{\text{Backup}}\right) \cdot m, \quad (3)$$

where p represents unit pricing, and q represents quantities.

For Amazon S3, costs include inbound storage and data transfer fees, calculated as follows:

$$C_{\text{S3}}(m) = \left(p_{\text{S3}} \cdot q_{\text{S3}} + p_{\text{DT}} \cdot q_{\text{DT}}\right) \cdot m. \quad (4)$$

ELB bases its cost on the volume of data the load balancer processes, typically measured in GB or TB per month. The cost of ELB is determined by the volume of data processed, represented as:

$$C_{\text{ELB}}(m) = p_{\text{ELB}} \cdot q_{\text{ELB}} \cdot m. \quad (5)$$

The total cost of operating in the cloud, C_{cloud}, over m months, is the sum of the expenses from Amazon FSx for Lustre, Amazon S3, and ELB, as defined in Equation (6).

$$C_{\text{cloud}}(m) = C_{\text{FSx}}(m) + C_{\text{S3}}(m) + C_{\text{ELB}}(m). \quad (6)$$

5.2 On-premises HPC costs

The cost evaluation of on-premises solutions focuses on the total expenses required for deploying and operating computing infrastructure locally. These expenses represent the total cost of ownership, which comprises fixed costs, capital expenditures (CapEx), and variable costs, which account for operational expenditures (OpEx). The total costs are expressed as follows:

$$C_{\text{on-premises}} = C_{\text{CapEx}} + C_{\text{OpEx}} \tag{7}$$

Fixed costs (CapEx) are required to build the infrastructure and do not depend on its use over time. These include acquiring hardware components, networking equipment, storage systems, and facility preparation costs. Mathematically, fixed costs can be expressed as:

$$C_{\text{CapEx}} = N \cdot C_{\text{Servers}} + C_{\text{Network}} + C_{\text{Storage}} + C_{\text{Facilities}} \tag{8}$$

where N is the number of hardware units, C_{Servers}, C_{Network}, and C_{Storage} are the costs associated with individual hardware components, network equipment, and storage systems, respectively, and $C_{\text{Facilities}}$ represents the expenses for facility preparation. Facility costs can be excluded if the existing infrastructure can sufficiently host the equipment without modifications.

Variable costs (OpEx) reflect the operating costs incurred over the lifecycle of the system and scale with usage over time. This includes electricity consumption, maintenance, depreciation, software licenses, and personnel costs. Assuming consistent monthly usage, variable costs are modeled as follows:

$$C_{\text{OpEx}} = m \cdot (C_{\text{Electricity}} + C_{\text{Staff}} + C_{\text{Licenses}} + C_{\text{Maintenance}} + C_{\text{Depreciation}}) \tag{9}$$

where m denotes the system's operating time in months. Electricity consumption contributes significantly to variable costs, which are calculated as follows:

$$C_{\text{Electricity}}(h) = \frac{24 \cdot 365}{12 \cdot 1000} \cdot c_{\text{Electricity}} \cdot (N \cdot P_{\text{Servers}} + P_{\text{Network}} + P_{\text{Cooling}}) \tag{10}$$

Here, P_{Servers}, P_{Network}, and P_{Cooling} denote the power consumption of the respective components in Watts, and $c_{\text{Electricity}}$ is the electricity cost per kWh. For systems with power-saving modes such as idle or hibernate, the calculation can be refined to account for the proportion of time spent in each mode.

Maintenance costs, which include hardware repairs and replacements, are often modeled as a percentage of the total initial hardware investment:

$$C_{\text{Maintenance}} = r \cdot (N \cdot C_{\text{Servers}} + C_{\text{Network}} + C_{\text{Storage}}) \tag{11}$$

where r represents the annual maintenance rate. Personnel costs depend on the total working hours required for operation and maintenance, expressed as:

$$C_{\text{Staff}}(h) = h \cdot C_{\text{Staff/hour}} \tag{12}$$

where h is the total monthly working hours and $C_{\text{Staff/hour}}$ is the hourly wage. Software licensing costs C_{Licenses} account for recurring software expenses required for operation. This cost model provides a detailed breakdown of the financial components of deploying and maintaining on-premise solutions. It allows adaptation to specific local conditions such as hardware prices, electricity tariffs, and labor costs and provides a robust framework for evaluating cost-effectiveness compared to alternative computing solutions.

5.3 On-premises and cloud costs comparison

Cost models assess computing infrastructure's financial viability for comparing on-premises solutions and cloud services. These models perform detailed analyses and determine economical approaches for specific computational tasks. Budget, standard, and high-performance on-premise solutions are considered with different complexities and configurations. Table 7 shows the main parameters for calculating the cost of an on-premise solution.

As a cloud service provider, AWS was selected with the nearest data center regarding latency, located in Frankfurt, Germany. The cost analysis for the cloud solution was conducted using formulas (3)–(6) based on data obtained from the AWS calculator[6]. The cost estimation for the on-premises solution was performed using formulas (7)–(12), leveraging publicly available data on equipment specifications, facility expenses, electricity rates[7], and staff salaries[8]. The comparison was visualized through a month-by-month analysis following formulas (1), (2) and is shown in Fig. 7.

The evaluation shows that the on-premises solution incurs significantly higher costs in the initial months due to substantial CapEx. After the first year, the monthly on-premises solution costs exhibit a noticeable increase. This rise reflects the inclusion of annual depreciation and maintenance provisions, accounting for the lifecycle and potential replacement of hardware components. While this adjustment increases monthly costs, the on-premises solution remains more cost-effective than the cloud alternative beyond the 12-month.

In contrast, the cloud-based solution demonstrates a consistent and predictable cost structure, reflecting the subscription-based pricing model of providers such as AWS. This flat monthly cost remains advantageous for short-term projects or applications requiring rapid scalability and minimal initial investment. However, this fixed-cost model becomes less favorable for long-term high-performance workloads due to its inability to benefit from economies of scale or decreasing marginal costs.

The results confirm that on-premises solutions offer clear financial advantages for computationally intensive projects exceeding one year, particularly in regions like Armenia where operational costs—such as electricity, real estate, and labor—are relatively

[6] `https://calculator.aws`

[7] `https://energyagency.am/page_pdf/sakagner`

[8] `https://www.paylab.com/am/salaryinfo/information-technology/systems-administrator`

Table 7. Cost Parameters for On-Premises Configurations

Characteristics	**Budget**	**Standard**	**High**
Configuration Details	10 nodes, basic cooling	50 nodes, air cooling, 150TB storage	150 nodes, CRAH cooling, 600TB storage
Node Specifications	Intel i7 12700, 16GB ECC RAM, 480GB SSD, 1.2TB HDD	Intel Xeon E5-2660 v3, 64GB ECC RAM, 480GB SSD, 1.2TB HDD	Intel Xeon E5-2660 v3, 160GB ECC RAM, 2x480GB SSD, 2x1.2TB HDD, Intel X520 10GbE NIC
Cooling System	Basic air cooling	Air-cooled chiller	CRAH with VFD chiller/tower
C_{Node} (USD)	800	1,500	2,100
C_{Servers} (USD)	8,000	75,000	315,000
C_{Storage} (USD)	—	4,500	18,000
$C_{\text{Facilities}}$ (USD)	—	—	140,000
C_{Network} (USD)	—	—	31,500
$C_{\text{Maintenance}}$ (%)	—	—	5
$C_{\text{Depreciation}}$ (%)	—	—	10
C_{Staff} (USD/month)	—	—	$800 \times 3 = 2,400$
P_{Servers} (W)	200	320	495
P_{Cooling} (W)	2,400	26,250	110,000
P_{Network} (W)	—	—	2,250

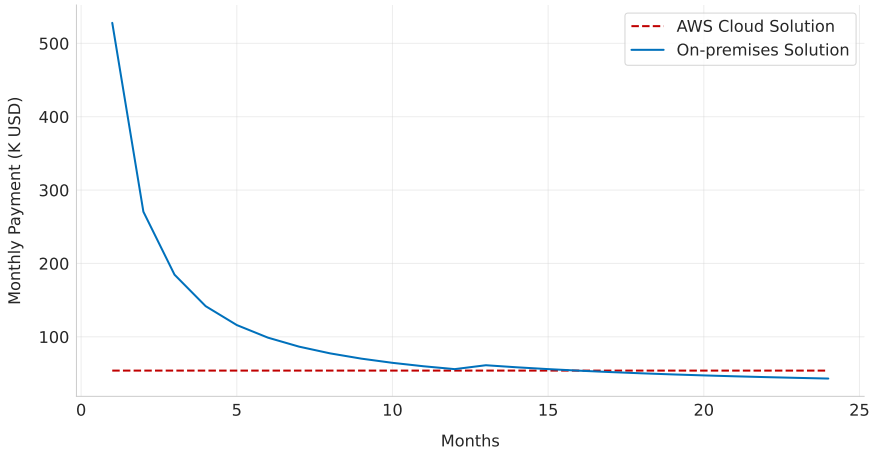


Fig. 7. On-Premises and Cloud Comparison

low. These results corroborate earlier studies, emphasizing the cost-efficiency of on-premises infrastructure in similar economic contexts. The lower costs of electricity and salaries in Armenia further amplify the financial feasibility of this approach and make it a more attractive option for local organizations or research institutions.

In summary, while cloud-based solutions remain suitable for short-term and highly variable computational needs, on-premises infrastructure demonstrates superior cost-effectiveness for long-term, resource-intensive projects. These results provide a tradeoff for decision-making in selecting computational infrastructure, particularly for institutions in regions with favorable economic conditions for on-premises deployment.

6 Conclusion and future work

As demand for cloud-based HPC continues to increase, existing market solutions often cannot meet the needs of large research projects due to high costs. As shown, on-premise solutions are more cost-effective than PAYG models. Despite the topic's relevance and extensive research, there are still hardly any viable alternatives. We have focused on three development strategies for such systems by comprehensively exploring various approaches.

Our research, which examines systems from both a file and object perspective, shows the convergence of Lustre parallel file systems with MinIO object storage as a promising solution. Specifically, as a COSBench-evaluated cloud storage solution, this converged approach performs better than the popular Ceph system, achieving over 20% better average response time and throughput metrics. As an HPC file system, Lustre delivers an average of 30% faster performance across all major operations. This performance advantage becomes much more noticeable with higher workloads.

The convergence of cloud technologies with HPC systems offers significant benefits such as improved scalability, improved data availability, and simplified workflow management. Although the initial setup is more complex compared to Ceph or JuiceFS, the convergence of Lustre with MinIO appears more promising.

Therefore, the convergence of Lustre with MinIO is the most compelling option for further investigation. Further exploration of this solution will contribute to developing an open-source HPC system with a fully integrated S3 gateway. The planned experiments will cover a broader range of workloads and compare the results with alternative object storage solutions, and utilize memory pooling technologies. Additionally, we will extend our experiments to various infrastructures, including serverless computing environments, containerized applications, virtual machines, traditional HPC setups, and clustered systems (Petrosyan and Astsatryan, 2022; Astsatryan et al., 2017, 2004).

Acknowledgements

The research was supported by the Science Committee of the Republic of Armenia by the project entitled "Self-organized Swarm of UAVs Smart Cloud Platform Equipped with Multi-agent Algorithms and Systems" (Nr. 21AG-1B052).

References

Astsatryan, H., Narsisian, W., Kocharyan, A., Da Costa, G., Hankel, A., Oleksiak, A. (2017). Energy optimization methodology for e-infrastructure providers, *Concurrency and Computation: Practice and Experience* **29**(10), e4073.

Astsatryan, H., Shoukourian, Y., Sahakyan, V. (2004). The armcluster project: brief introduction, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA'04*, pp. 1291–1295.

Chen, H.-B., Grider, G., Montoya, D. R. (2017). An early functional and performance experiment of the marfs hybrid storage ecosystem, *2017 IEEE International Conference on Cloud Engineering (IC2E)*, IEEE, pp. 59–66.

Chen, H.-M., Li, C.-J., Ke, B.-S. (2017). Designing a simple storage services (s3) compatible system based on ceph software-defined storage system, *Proceedings of the 2017 2nd International Conference on Multimedia Systems and Signal Processing*, pp. 6–10.

Cho, K.-J., Kang, I., Kim, J.-S. (2023). Arkfs: A distributed file system on object storage for archiving data in hpc environment, *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, IEEE, pp. 301–311.

Dai, D., Gatla, O. R., Zheng, M. (2019). A performance study of lustre file system checker: Bottlenecks and potentials, *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*, IEEE, pp. 7–13.

Duplyakin, D., Ricci, R., Maricq, A., Wong, G., Duerig, J., Eide, E., Stoller, L., Hibler, M., Johnson, D., Webb, K. et al. (2019). The design and operation of {CloudLab}, *2019 USENIX annual technical conference (USENIX ATC 19)*, pp. 1–14.

Durner, D., Leis, V., Neumann, T. (2023). Exploiting cloud object storage for high-performance analytics, *Proceedings of the VLDB Endowment* **16**(11), 2769–2782.

Emeras, J., Besseron, X., Varrette, S., Bouvry, P., Peters, B. (2016). Hpc or the cloud: a cost study over an xdem simulation, *Proc. of the 7th International Supercomputing Conference in Mexico (ISUM 2016). Puebla, México*.

Gadban, F. (2022). *Analyzing Convergence Opportunities of HPC and Cloud for Data Intensive Science*, PhD thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky.

Gadban, F., Kunkel, J. (2021). Analyzing the performance of the s3 object storage api for hpc workloads, *Applied Sciences* **11**(18), 8540.

Gadban, F., Kunkel, J., Ludwig, T. (2020). Investigating the overhead of the rest protocol when using cloud services for hpc storage, *High Performance Computing: ISC High Performance 2020 International Workshops, Frankfurt, Germany, June 21–25, 2020, Revised Selected Papers 35*, Springer, pp. 161–176.

Gudu, D., Hardt, M., Streit, A. (2014). Evaluating the performance and scalability of the ceph distributed storage system, *2014 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 177–182.

Hennecke, M. (2020). Daos: A scale-out high performance storage stack for storage class memory, *Supercomputing frontiers* **40**.

Huang, W.-C., Lai, C.-C., Lin, C.-A., Liu, C.-M. (2015). File system allocation in cloud storage services with glusterfs and lustre, *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 1167–1170.

Inman, J. T., Vining, W. F., Ransom, G. W., Grider, G. A. (2017). Marfs, a near-posix interface to cloud objects, *; Login* **42**(LA-UR-16-28720; LA-UR-16-28952).

Jeong, K., Duffy, C., Kim, J.-S., Lee, J. (2019). Optimizing the ceph distributed file system for high performance computing, *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 446–451.

Jones, M., Kepner, J., Arcand, W., Bestor, D., Bergeron, B., Gadepally, V., Houle, M., Hubbell, M., Michaleas, P., Prout, A. et al. (2017). Performance measurements of supercomputing and cloud storage solutions, *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, pp. 1–5.

Lackschewitz, N. M., Krey, S., Nolte, H., Christgau, S., Oeste, S., Kunkel, J. (2022). Performance evaluation of object storages, *NHR2022* .

Li, H., Zhang, S., Guo, Z., Huang, Z., Qian, L. (2020). Test and optimization of large-scale ceph system, *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, IEEE, pp. 237–241.

Lillaney, K., Tarasov, V., Pease, D., Burns, R. (2019). Towards marrying files to objects, *arXiv preprint arXiv:1908.11780* .

Liu, J., Koziol, Q., Butler, G. F., Fortner, N., Chaarawi, M., Tang, H., Byna, S., Lockwood, G. K., Cheema, R., Kallback-Rose, K. A., Hazen, D., Prabhat, M. (2018). Evaluation of hpc application i/o on object storage systems, *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*, pp. 24–34.

Lockwood, G. K., Snyder, S., Wang, T., Byna, S., Carns, P., Wright, N. J. (2018). A year in the life of a parallel file system, *SC18: International conference for high performance computing, networking, storage and analysis*, IEEE, pp. 931–943.

Lofstead, J., Jimenez, I., Maltzahn, C., Koziol, Q., Bent, J., Barton, E. (2016). Daos and friends: a proposal for an exascale storage system, *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, pp. 585–596.

Luettgau, J., Martinez, H., Olaya, P., Scorzelli, G., Tarcea, G., Lofstead, J., Kirkpatrick, C., Pascucci, V., Taufer, M. (2023). Nsdf-services: Integrating networking, storage, and computing services into a testbed for democratization of data delivery, *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, pp. 1–10.

Manubens, N., Smart, S. D., Quintino, T., Jackson, A. (2022). Performance comparison of daos and lustre for object data storage approaches, *2022 IEEE/ACM International Parallel Data Systems Workshop (PDSW)*, IEEE, pp. 7–12.

Milojicic, D., Faraboschi, P., Dube, N., Roweth, D. (2021). Future of hpc: Diversifying heterogeneity, *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 276–281.

Palankar, M. R., Iamnitchi, A., Ripeanu, M., Garfinkel, S. (2008). Amazon s3 for science grids: a viable solution?, *Proceedings of the 2008 international workshop on Data-aware distributed computing*, pp. 55–64.

Panda, D. K., Lu, X., Shankar, D. (2022). *High-performance big data computing*, MIT Press, London.

Petrosyan, D., Astsatryan, H. (2022). Serverless high-performance computing over cloud, *Cybernetics and Information Technologies* **22**(3), 82–92.

Schwan, P. et al. (2003). Lustre: Building a file system for 1000-node clusters, *Proceedings of the 2003 Linux symposium*, Vol. 2003, pp. 380–386.

Smith, H. (2016). *Data Center Storage: Cost-Effective Strategies, Implementation, and Management*, CRC Press, Boca Raton, London, New York.

Smith, P., Harrell, S. L., Younts, A., Zhu, X. (2019). Community clusters or the cloud: Continuing cost assessment of on-premises and cloud hpc in higher education, *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, Association for Computing Machinery, pp. 1–4.

Spiga, D., Ciangottini, D., Costantini, A., Cutini, S., Duma, C., Gasparetto, J., Lubrano, P., Martelli, B., Ronchieri, E., Salomoni, D. et al. (2022). Open-source and cloud-native solutions for managing and analyzing heterogeneous and sensitive clinical data, *International Symposium on Grids and Clouds 2022, ISGC 2022*.

Van der Ster, D., Wiebalck, A. (2014). Building an organic block storage service at cern with ceph, *Journal of Physics: Conference Series*, Vol. 513, IOP Publishing, p. 042047.

Weil, S. A., Brandt, S. A., Miller, E. L., Maltzahn, C. (2006). Crush: Controlled, scalable, decentralized placement of replicated data, *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, pp. 122–es.

Weil, S., Brandt, S. A., Miller, E. L., Long, D. D., Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system, *Proceedings of the 7th Conference on Operating Systems Design and Implementation (OSDI'06)*, pp. 307–320.

Zhang, X., Wang, Y., Wang, Q., Zhao, X. (2019). A new approach to double i/o performance for ceph distributed file system in cloud computing, *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, IEEE, pp. 68–75.

Zheng, Q., Chen, H., Wang, Y., Duan, J., Huang, Z. (2012). Cosbench: A benchmark tool for cloud object storage services, *2012 IEEE Fifth International Conference on Cloud Computing*, IEEE, pp. 998–999.

Received December 9, 2024 , revised February 21, 2025, accepted February 21, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 200–220
https://doi.org/10.22364/bjmc.2025.13.1.11

Multi-Method Simulation and Optimisation for Maximising Benefits in Renewable Energy Communities: A Real-World Case Study from Italy

Stefano SANFILIPPO[1] ⋆, Lorenzo FARINA[2] ⋆⋆, Pietro DE VITO[1] ⋆⋆⋆, José Juan HERNÁNDEZ-CABRERA[3] †, José Juan HERNÁNDEZ-GÁLVEZ[3] ‡, José ÉVORA-GÓMEZ[3] §

[1] STAM S.r.l., Genova, Italy
[2] Department of Informatics, Bioengineering, Robotics and Systems Engineering - University of Genova, Genova, Italy
[3] Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería - Universidad de Las Palmas de Gran Canaria, Las Palmas, Gran Canaria, Spain

Abstract. This paper introduces a novel multi-method modelling framework for Renewable Energy Communities (RECs), integrating agent-based modelling, discrete-event simulation, and system dynamics. This hybrid approach enables a comprehensive assessment of RECs, capturing both their technical and economic dynamics. The work's key contributions are twofold: (i) a flexible technical modelling framework adaptable to diverse geographical and regulatory contexts, and (ii) an advanced optimisation model aimed at minimising costs and maximising benefits for decision support.

The optimisation model has been built upon the modelling framework and can be adjusted to various REC configurations, allowing for variations in photovoltaic capacity, demand patterns, energy price structures, and regulatory schemes. This flexibility enables a policy-aware and context-sensitive simulation and optimisation of REC operations. The model enables the evaluation of a wide range of scenarios, helping stakeholders assess both short-term and long-term technical and economic performance, making it a robust tool for forecasting and strategic planning.

⋆ s.sanfilippo@stamtech.com ORCID: 0009-0001-0547-6222
⋆⋆ lorenzo.farina@edu.unige.it ORCID: 0009-0005-5159-1861
⋆⋆⋆ p.devito@stamtech.com ORCID: 0000-0002-7353-2011
† josejuan.hernandez@ulpgc.es ORCID: 0000-0003-2427-2441
‡ jose.galvez@ulpgc.es ORCID: 0009-0008-3626-7520
§ jose.evora@ulpgc.es ORCID: 0000-0001-9348-7265

A real-world case study in Val d'Aosta, Italy, demonstrates the model's applicability and effectiveness. The study highlights the framework's ability to incorporate country-specific REC regulations while optimizing REC configurations. Results show a reduction in external energy reliance and an increase in shared energy, leading to enhanced energy autonomy and economic benefits. These findings validate the model's robustness and scalability, establishing it as a pioneering framework for REC planning and policy innovation.

Keywords: Renewable Energy Communities, Modelling Approach, Multi-method Simulation, Agent-based modelling, Discrete-event Simulation, System Dynamics, Meta-heuristic Optimisation, Flexibility, Replicability, Scalability, Economic Benefits

1 Introduction

The Clean Energy for All Europeans package, introduced by the European Commission in November 2016 (Capros et al., 2018), includes key directives such as the Renewable Energy Directive (RED II) and the Internal Market for Electricity. These directives aim to promote distributed energy generation, enhance the role of Renewable Energy Sources (RES) (Frieden et al., 2020), and empower citizens as active participants in the energy transition (Sokolowski, 2018). As a result, new energy management systems have emerged, including microgrids (Sanfilippo and et al., 2023), smart cities (Zacepins et al., 2019), and Renewable Energy Communities (RECs).

A REC is a legally recognized entity that operates independently while complying with national regulations. It is characterized by open and voluntary participation, with governance controlled by its members—individuals, small and medium-sized enterprises, or local authorities such as municipalities. Unlike traditional energy entities, RECs prioritize environmental, economic, and social benefits over financial profit (Vetrò and Brignoli, 2025; Lode et al., 2022).

Figure 1 illustrates the general concept of a REC, which can include different types of participants. Pure producers, represented by photovoltaic (PV) panels, generate electricity that is shared among community members. Consumers, depicted as individual houses, rely on this shared energy for their needs. Additionally, prosumers—buildings equipped with rooftop PV systems—can generate electricity locally. These prosumers have the flexibility to consume the energy they produce, share surplus electricity with other REC members, or feed excess power into the main grid, contributing to a more balanced and efficient energy distribution within the community.

The adoption of RECs is accelerating across Europe, supported by regulatory frameworks and financial incentives. A recent study identified nearly 4,000 energy communities in the EU, involving approximately 900,000 members, with Germany, the Netherlands, and Denmark leading the movement (Koltunov et al., 2023). Figure 2 presents an overview of the transposition status of REC definitions in different European countries.

However, the widespread deployment of RECs, combined with the increasing integration of variable renewable energy sources, presents challenges for efficient and sustainable electricity grid management. Among the different RES technologies adopted within RECs, PV systems are the most prevalent due to their modularity, cost-effectiveness, and ease of integration into buildings and ground installations (Gianaroli et al., 2024).

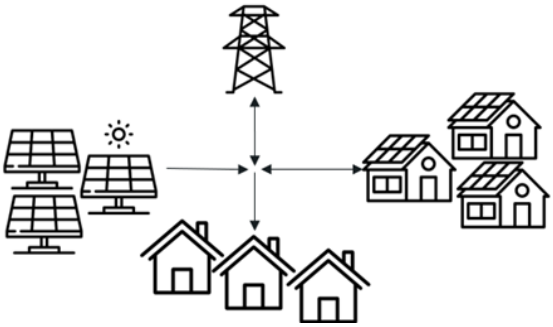


Fig. 1. Renewable Energy Community concept.

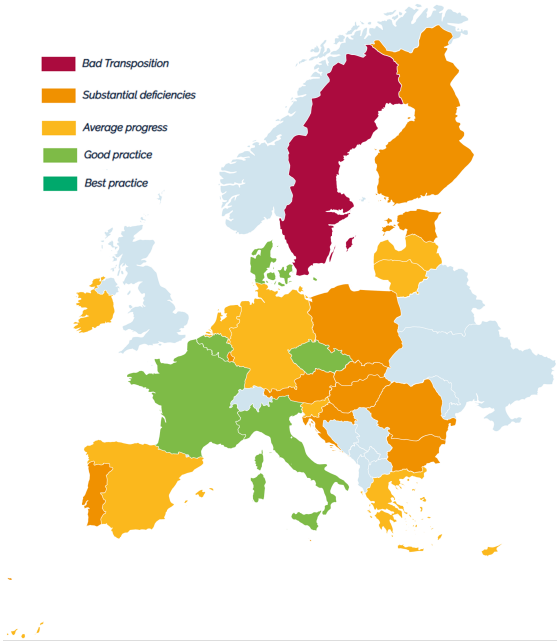


Fig. 2. Transposition map of REC and Citizen Energy Community definitions – April 2024, sourced from (REScoop Website, 2024).

Despite the rapid growth of RECs, there is a need for advanced optimisation strategies to enhance their economic, environmental, and operational performance. Several business models for RECs have been proposed, differing in governance structures, pricing mechanisms, and energy distribution strategies (Barabino et al., 2023). Furthermore, the complexity of REC management necessitates sophisticated optimisation approaches that account for diverse technologies, including PV systems, energy storage, electric vehicles, and heating/cooling systems. Existing studies have addressed optimisation from two key perspectives:

- Real-time energy management considers optimisation methods that support real-time control of energy assets to minimise costs, enhance flexibility, and ensure grid stability. These approaches include demand response programs, local flexibility markets, and multi-objective energy management frameworks (Cruz-De-Jesús et al., 2023; Caliano et al., 2022).
- Strategic planning and system design encorporating optimisation techniques which aid in the long-term planning of RECs by determining optimal infrastructure deployment and energy-sharing mechanisms. Notably, mathematical models incorporating geographical, meteorological, and demographic data have been developed to optimize PV installation and maximise economic and environmental benefits (Orlando et al., 2023; Lazzari et al., 2023).

From an operational perspective, REC members — producers, consumers, and prosumers — can be modeled as agents within a distributed multi-agent system (Listopad, 2019). Additionally, dynamic system modelling is crucial for simulating complex interactions among energy assets (Mihailovs and Cakula, 2020). In this regard, recent research has explored methods for assessing REC feasibility under economic and regulatory uncertainty (Pagnini et al., 2024; Cutore et al., 2023).

As shown, various works proposed modelling and optimisation approaches in the context of RES optimal design for REC application. Mathematical modelling in REC, considering the optimisation, is used to maximise profit based on the number of prosumers and consumers, finding the profitability by comparing the optimal REC and without REC case (Sassone et al., 2024) however, the optimisation models, do not consider the overall useful time of the investment, as in this work.

2 Objectives

This research is driven by the objectives and broader framework of the PROBONO project (PROBONO Project, 2022), which aims to foster the development of sustainable, energy-positive, and zero-carbon Green Building Neighborhoods across Europe. The PROBONO project, funded by the European Union's Horizon 2020 program, focuses on aligning the interaction between buildings, communities, and stakeholders, leveraging digitalisation and smart technologies to achieve net-zero emissions and energy-positive environments.

In this context, the goals of PROBONO closely align with the challenges faced by RECs, particularly in the transition to renewable energy. One of the critical issues in this

transition is optimising the performance of decentralized energy production systems, especially PV installations. Achieving these objectives involves not only determining the optimal size and configuration of PV systems — considering factors such as fluctuating energy demand, variable PV output, and differing regulatory frameworks and incentives — but also balancing capital expenditures (CAPEX) and operational expenditures (OPEX) with the economic benefits of reducing electricity purchases and increasing energy sales within the REC. Additionally, navigating the complex landscape of local regulations and incentive schemes, which can greatly impact the economic viability of RECs, presents further challenges. Without adaptable models, stakeholders often lack the tools needed to make informed decisions that consider the unique legal and economic conditions of their specific locations.

Optimally sizing PV systems not only enhances economic performance but also minimises excess electricity production, reducing the need to export surplus power to the grid. This improves self-consumption and resilience while helping to mitigate grid congestion by preventing overproduction, which can strain local distribution infrastructure. When PV systems are tailored to match energy demands and grid capacity, they boost energy efficiency and contribute to a more stable and reliable electrical grid.

The primary objective of this research is to develop a model that can be used both in a simulation and optimisation tool for RECs. This model is designed to help stakeholders make informed decisions regarding PV system sizing and energy management strategies, ultimately supporting the growth and success of sustainable energy communities. By optimising the performance of RECs, this research directly aligns with PROBONO's mission to demonstrate how Green Building Neighborhoods can achieve energy-positive outcomes while benefiting both society and the environment through innovative, decentralized energy solutions.

The model requires to be adaptable and replicable to various locations and scalable to different scenarios, if relevant data regarding PV production, budget requirements, electricity consumption, and local regulations related to shared energy incentives are available. Through its application, the model will provide valuable insights into how the integration of REC within local communities can be optimised to enhance both economic and environmental outcomes.

3 Proposed Approach

This section introduces a model for RECs designed to conduct comprehensive techno-economic analyses and optimise their benefits. This work builds upon and extends the findings presented in the conference paper delivered at the 15th Conference on Data Analysis Methods for Software Systems in November 2024. A REC is a complex system, that requires dynamic management of various interconnected components, including buildings and their production and consumption. To address this complexity, a multi-method modelling approach is adopted, combining Agent-Based Modelling (ABM), Discrete Event Simulation (DES), and System Dynamics (SD):

- Each PoD is modelled as an agent with specific attributes, such as installed renewable energy systems, electricity generation and electricity demand, and the ability

to inject or withdraw electricity from the grid. ABM allows for the detailed representation of individual behaviours and decision-making processes, capturing their interactions and contributions to the REC.
- DES considers the dynamic interactions and adjustments among PoDs at each time step, allowing the system to adapt to real-time changes, such as fluctuations in energy demand, production, or grid conditions.
- At the community level, SD synthesizes the results from individual agents, treating the REC as an interconnected system. This approach captures aggregated trends (e.g., total energy shared, overproduction, overconsumption) and long-term economic impacts on the REC.

This model serves dual purposes of simulation and optimisation, offering a robust framework for analysing and enhancing REC performance over its entire lifetime. It ensures that the proposed solutions are not only effective in the short term but also sustainable and economically viable over time.

The simulation lays the foundation for the optimisation process, which determines the optimal configuration of nominal PV capacities at each PoD. It analyses electricity exchanges at both PoD and REC levels, evaluates financial metrics such as costs, revenues, incentives, and net benefits under both REC and non-REC scenarios, and identifies the optimal sizing of PV systems to maximise economic and environmental benefits. The primary objective is to minimise costs while ensuring uninterrupted electricity demand coverage within the REC.

4 Model Definition

The aim of the model is to capture the state within a REC consisting of n PoDs in a generic area. A PoD is a specific location within a REC where electricity is either consumed, generated, or both. It represents a physical or virtual point in the electrical grid where energy flow is monitored and managed. It typically corresponds to a building, household, or any unit equipped with its own electricity meter, which can either consume electricity from the grid or supply electricity back to it, using a PV system.

The energy demand of the *i-th* PoD, $e_i^D(t)$, must be determined for each time period t. In addition, the PV production at the *i-th* PoD during each time period t, $e_i^P(t)$, is calculated by multiplying the PV production profile for 1 kW, $\phi(t)$, by the nominal power of the PV system installed at that PoD, P_i^{PV} as shown in equation 1.

$$e_i^P(t) = \phi(t) \times P_i^{PV} \tag{1}$$

The net energy exchange, $e_i^X(t)$, represents the net energy flow at PoD i during a specific time period t. It is calculated as the difference between the energy produced, $e_i^P(t)$, and the energy demanded, $e_i^D(t)$, at the same PoD and time interval, as expressed in equation 2.

$$e_i^X(t) = e_i^P(t) - e_i^D(t) \tag{2}$$

The energy exchange $e_i^X(t)$ at the *i-th* PoD during a given time period t can either result in energy being injected into the grid or withdrawn from it. The energy injected,

denoted as $e_i^I(t)$, represents the surplus energy exported to the grid when the production exceeds the demand. Conversely, the energy withdrawn, denoted as $e_i^W(t)$, represents the energy imported from the grid to cover the demand when it exceeds the production.

If $e_i^X(t)$ is positive, it is considered as energy injected, $e_i^I(t)$, and $e_i^W(t)$ is set to zero. Conversely, if $e_i^X(t)$ is negative, it is considered as energy withdrawn, $e_i^W(t)$, and $e_i^I(t)$ is set to zero. This ensures that, for any given time period t, the energy exchange is exclusively categorized as either injected or withdrawn. Mathematically, this can be expressed as shown in equation 3.

$$e_i^I(t) = \max(e_i^X(t), 0) \qquad e_i^W(t) = \max(-e_i^X(t), 0) \tag{3}$$

The energy shared by the *i-th* PoD, $\bar{e}_i(t)$, during a given period corresponds to the energy made available from the PoD i for the REC. This is calculated as shown in equation 4.

$$\bar{e}_i(t) = \min(e_i^I(t),\, e_i^W(t)) \tag{4}$$

This equation is essential because it quantifies the actual contribution of each PoD to the shared energy pool of the REC. By using the minimum between the energy injected and the energy withdrawn, the model ensures that only the surplus energy, which is effectively available for sharing, is accounted for. This avoids overestimating the shared energy, as it limits the contribution to the actual availability of energy at the PoD. This calculation is critical for balancing the energy flows within the REC. It determines the total energy shared, directly influencing the financial benefits derived from shared incentives or reduced energy costs. Furthermore, it ensures the equitable distribution of shared energy across all PoDs, aligning individual contributions with the overall objectives of the REC. By accurately representing the energy available for sharing, this equation promotes efficient energy management and supports the collective operation of the REC. It also serves as a foundational component for aggregating and analysing energy contributions across all PoDs in the REC.

By aggregating the energy contributions from all PoDs, the REC quantifies the total energy shared and exchanged during a period t. This includes the aggregated energy injected into the grid, withdrawn from the grid, and shared within the REC. The total values are calculated by summing the respective energy components across the number of PoDs n, as represented in equation 5.

$$e^I(t) = \sum_{i=1}^{n} e_i^I(t), \quad e^W(t) = \sum_{i=1}^{n} e_i^W(t), \quad \bar{e}(t) = \sum_{i=1}^{n} \bar{e}_i(t) \tag{5}$$

This aggregated perspective allows the REC to evaluate its collective energy performance, effectively balancing energy flows between injection, withdrawal, and sharing. It provides a comprehensive view of the energy dynamics within the community, serving as a foundation for accurate accounting and reporting of energy exchanges.

The financial transactions associated with energy exchanges in the REC include both the revenue from selling injected energy and the cost of buying withdrawn energy. The revenue from selling the injected energy, denoted as $r(t)$, is calculated by multiplying the energy injected into the grid, $e^I(t)$, by the market sell price at time t, $p^S(t)$.

Similarly, the cost of buying the withdrawn energy, denoted as $c(t)$, is calculated by multiplying the energy withdrawn from the grid, $e^W(t)$, by the purchase price at time t, $p^B(t)$. These relationships are expressed in equations 6.

$$r(t) = e^I(t) \times p^S(t) \qquad c(t) = e^W(t) \times p^B(t) \tag{6}$$

The equations formalized so far express the dynamics of energy and financial exchanges as continuous functions of time, enabling detailed analysis at any specific moment within the study duration. However, for practical and comparative purposes, these time-dependent equations need to be annualized, in order to align the analysis with standard accounting practices. This ensures that the CAPEX and the OPEX are appropriately accounted for the financial statements. Specifically, CAPEX is reflected in the balance sheet as an asset and is typically amortized or depreciated over its useful life. In contrast, OPEX is included in the income statement, representing the periodic operational costs incurred during the study duration. This involves aggregating the results over each year of the study to obtain yearly metrics such as total energy injected, withdrawn, or shared, as well as the corresponding revenues and costs.

The total energy demanded, injected and withdrawn by the REC during a given year y, denoted as E_y^D, E_y^I and E_y^W respectively, is obtained by summing the energy values over all periods within that year. This is represented mathematically in equations 7.

$$E_y^D = \int_y e^D(t)\,dt \qquad E_y^I = \int_y e^I(t)\,dt \qquad E_y^W = \int_y e^W(t)\,dt \tag{7}$$

The revenue from selling injected energy and the costs for buying withdrawn energy at the REC level during a specific year y are obtained by integrating the respective instantaneous values over the year. These are expressed in equations 8.

$$R_y = \int_y r(t)\,dt \qquad C_y = \int_y c(t)\,dt \tag{8}$$

The capital expenditures for each PoD, $CAPEX_i$, is computed individually, taking into account the cost of the PV system per kW, C_{kW}^{PV}, multiplied by the installed PV power in the *i-th* PoD as shown in equation 9.

$$CAPEX_i = C_{kW}^{PV} \times P_i^{PV} \tag{9}$$

The cumulative capital expenditure of the REC is aggregated across all PoDs. The total *CAPEX* represents the sum of the individual capital expenditures of each PoD, as shown in equation 10:

$$CAPEX = \sum_{i=1}^{n} CAPEX_i \tag{10}$$

$OPEX_y$, represent the yearly costs incurred to maintain and operate the REC, including maintenance, administrative expenses, and other operational fees. The yearly $OPEX_y$ is calculated as a percentage f of the total capital expenditure (*CAPEX*) for each PoD, aggregated across all PoDs, as shown in equation 11.

$$OPEX_y = f \cdot CAPEX \tag{11}$$

The yearly cash-flow, denoted as F_y, represents the net financial balance of the REC during year y, incorporating several factors: operating expenditures $OPEX_y$, the cost of energy purchased C_y, the revenue generated from selling injected energy R_y, and any incentives from the country to the REC I_y. This is given by equation 12:

$$F_y = OPEX_y + C_y - R_y - I_y \tag{12}$$

The term $OPEX_y$ includes the yearly operational costs required to maintain and manage the REC, such as maintenance and administrative expenses. The cost of energy purchased, C_y, accounts for the expenses incurred from withdrawing energy from the grid to satisfy the REC's energy demand. Revenue, R_y, represents the income generated by selling surplus energy injected into the grid, benefiting from market prices or other mechanisms. State incentives, I_y, are financial supports or subsidies provided annually by governmental or local authorities. These incentives are typically tied to renewable energy policies, aiming to encourage the adoption of sustainable practices. They may vary depending on the country or region and are calculated based on the REC's operational characteristics, such as the amount of energy shared or injected into the grid, or compliance with specific regulatory requirements. By incorporating I_y, the cash-flow model reflects both market-based earnings and additional policy-driven benefits.

Finally, the total costs with REC, denoted as T, are calculated as the sum of the initial investment *CAPEX* and the discounted yearly cash flows F_y over the project lifespan Y. This is expressed in equation 13:

$$T = CAPEX + \sum_{y=1}^{Y} F_y \cdot D_y \tag{13}$$

Here, all cash flows are adjusted to their present value using the discount factor D_y, ensuring that the financial evaluation reflects the time value of money over the REC's lifetime. The yearly discount rate, D_y, is used to account for the time value of money in year y, ensuring that cash flows are appropriately adjusted to their present value. The adjusted yearly discount rate is computed as shown in equation 14. d represents the annual discount rate, which reflects the cost of capital.

$$D_y = \frac{1}{(1+d)^y} \tag{14}$$

Now that the model for calculating costs with REC has been established, comparing these costs with those without REC allows for the evaluation of the economic benefit of implementing PV installations and constituting the REC.

The yearly costs without REC, T'_y, are determined by integrating the energy demand $e^D(t)$ multiplied by the price of energy $p^B(t)$, as calculated in equation 15.

$$T'_y = \int_y e^D(t) \cdot p^B(t)\, dt \tag{15}$$

The total costs in the scenario without REC, denoted as T', are obtained by summing the discounted yearly costs T'_y over the entire study duration. The discount factor D_y is applied to adjust each year's costs to their present value. This is expressed in equation 16.

$$T' = \sum_{y=1}^{Y} T'_y \cdot D_y \tag{16}$$

By comparing the costs with REC and the costs without REC, the economic benefit of implementing PV installations and constituting the REC can be evaluated. This comparison highlights the financial advantage provided by reduced operational costs and the potential earnings from energy sharing in the REC scenario.

5 Simulation

The mathematical model defines the energy and financial dynamics of the REC, providing a theoretical support for analysing its performance. This model has been implemented and further developed in the AnyLogic Software Platform (AnyLogic Company, 2024) to create a functional simulation environment. AnyLogic allows for the integration of different simulation paradigms within a single tool, such as Agent-Based, Discrete Event, and System Dynamics. Beyond its modelling capabilities, AnyLogic is also favored for its problem-solving orientation.

The simulation enables the practical application of the model, allowing for detailed analyses of REC behaviour under various scenarios. The tool aims to provide a time-based simulation, allowing for a detailed analysis of the REC's behaviour on an hourly basis. Each PoD has its own PV system and electricity demand, with only the electricity aspect of the REC being considered for simplicity.

The simulation operates with a time period t set to hourly intervals, reflecting the granularity of the input data. This hourly resolution ensures the accurate representation of fluctuations in electricity demand, PV production, and market prices. AnyLogic's flexibility has been utilized to integrate ABM and DES, effectively capturing the interactions within the REC. Each PoD is modelled as an agent, with attributes such as installed PV capacity, hourly electricity demand, and interactions with the REC and the grid.

The model is customisable by the user through an interface and an Excel file, providing flexibility to adjust the REC's structure and the characteristics of each PoD. In the simulation, every PoD is modelled as an agent, with its electricity demand specified hourly for an entire year sourced from an input Excel file. The Excel file is structured across multiple sheets. The first sheet contains information about the PoDs, with each row representing a PoD and providing details such as a unique identifier, year of construction, and available surface area for PV installation. The second sheet contains

columns for each consumer, with each column being the electricity demand of the corresponding PoD. This setup offers users flexibility by enabling them to dynamically set the composition of their REC and the electricity demand of each PoD before running the simulation..

During the simulation, the model captures key energy flows, including energy demand, $e_i^D(t)$, energy injected into the grid, $e_i^I(t)$, energy withdrawn from the grid, $e_i^W(t)$, and energy shared within the REC, $\bar{e}_i(t)$, on an hourly basis. They are calculated using AnyLogic's time-based modelling capabilities. Event-driven logic handles updates to external factors, such as changes in market prices or policy incentives, ensuring the simulation remains responsive to dynamic conditions.

When the simulation is executed, the model transfers all input data from the Excel file to its internal datasets and variables. For each PoD, this includes unique characteristics, hourly electricity demand, and hourly PV production for a unitary 1 kW installation. The electricity production of each PoD is calculated based on its installed PV capacity, P_i^{PV}, and the PV production profile, $\phi(t)$, which can either be provided as part of the input data or obtained dynamically via the PVGIS API. If the PVGIS API is used, the profile is based on specific assumptions such as polycrystalline panels, 14% of losses, and historical data from 2005 to 2020, averaged to simulate a generalized year.

The simulation operates for an entire year (8,784 hours to accommodate leap years, or 8,760 hours for non-leap years), providing results that can be extrapolated over the REC's lifetime for comprehensive investment analysis.

Outputs from the simulation include aggregated energy metrics, yearly financial performance indicators, and optimisation results for PV capacities. These outputs are stored in AnyLogic's datasets and visualized through its built-in analytics tools. The model's outputs include yearly energy metrics, such as total energy injected, E_y^I, withdrawn, E_y^W, and shared, $\bar{E}_y$, as well as financial metrics like yearly cash-flow, F_y, cumulative capital expenditure (*CAPEX*), and operating expenditure ($OPEX_y$). Additionally, the model provides insights into the REC's overconsumption, overproduction, and overall energy sharing behaviour.

The simulation results form the basis for optimising the nominal power of each PV system in the REC. By analysing the interplay between incentives, costs, and energy production, the model supports decision-making aimed at maximising financial incentives while minimising operational costs, ensuring the economic and energy efficiency of the REC.

6 Optimisation

For the optimisation, the AnyLogic experiment employs advanced meta-heuristics and the OptQuest solver (OptQuest Website, 2024) to perform optimisation procedures, leveraging AnyLogic's optimisation features. This procedure is specifically designed to handle complex systems with numerous decision variables that present analytical optimisation challenges.

Built on the simulation, the optimisation aims at determining the optimal configuration of installed PV at the PoD level. The optimisation seeks to maximise REC benefits

by finding the best combination of installed PV that balances production, demand, and energy sharing within the REC. Key objectives include:

The optimisation problem is mathematically formulated as:

$$\max_{x} [T' - T(x)] \tag{17}$$

$$\textit{with } x = P_i^{PV} \quad \forall i = 1, \ldots, n \tag{18}$$

The optimisation aims to maximise the economic benefit of the REC by minimising the total costs associated with REC operations $T(x)$ relative to the costs in a non-REC scenario T'. This approach provides a decision-support tool for determining the optimal PV capacity at each PoD to balance investment costs, energy flows, and financial incentives while achieving maximum savings.

The optimisation begins with initializing the input data, which includes:

- **Energy-related data**: Hourly electricity demand profiles and hourly PV production per kW at the PoD site.
- **Economic data**: Hourly energy prices, hourly purchasing costs, and other relevant financial parameters.

The input parameters, specifically the range of PV capacities, are also defined. The minimum value is set to zero (representing no installation), and the maximum value corresponds to the highest feasible capacity based on physical constraints.

The optimisation employs advanced meta-heuristics and the OptQuest solver (OptQuest Website, 2024) to perform an optimisation process based on AnyLogic's optimisation feature. The optimisation layer operates atop the simulation iteratively. The objective function, expressed in equation 17, is analogous to the Net Present Value (*NPV*) of the investment, with a notable enhancement: unlike traditional *NPV* calculations that consider only *CAPEX* as the first-year cash flow, the proposed formulation incorporates both revenues and *OPEX* in the first year. This approach reflects the assumption that installations occur at the beginning of the investment period.

The meta-heuristic optimisation layer addresses the complexity of the problem, characterized by a large number of decision variables and interdependencies. The process is visualized in Figure 3, which depicts the flowchart of the optimisation steps. The steps are:

1. **Simulation Execution**: The simulation runs using initialized input parameters and data, calculating the REC Benefits.
2. **Evaluate Stopping Criteria**: The optimisation process assesses whether the predefined stopping criteria—such as reaching the maximum number of iterations, achieving convergence thresholds, or applying a budget constraint to exclude financially unfeasible solutions and ensure alignment with the investor's capacity—have been satisfied. If these conditions are met, the process terminates and determines the optimal sizes for the installed PV panels. Otherwise, it updates the input parameters and continues.
3. **Update Input Parameters**: PV capacities are iteratively adjusted within the defined range to refine the solution and improve the objective function.

4. **Optimal Design Identification**: The solution yielding the maximum REC Benefits is selected, ensuring feasibility and adherence to constraints such as continuous electricity demand coverage and financial limits.

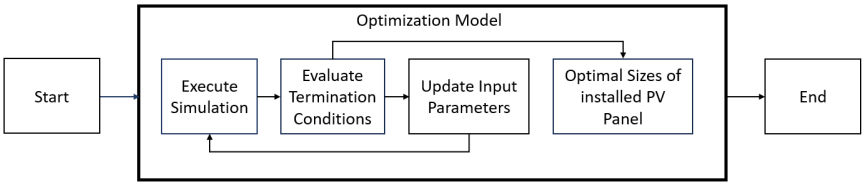


Fig. 3. Flowchart depicting the optimisation process.

This dual-layer optimisation offers flexibility for addressing both economic and operational objectives in REC design. By integrating simulation with meta-heuristic optimisation, the methodology ensures a robust and adaptable solution tailored to the specific characteristics and constraints of REC. The proposed approach is particularly suited for evaluating and optimising REC performance over its lifetime, delivering practical insights for stakeholders and policymakers.

7 Experimental Work

The case study is situated in the Aosta Valley, Italy, a region that benefits from the well-defined national regulations and incentives for renewable shared energy (Gianaroli et al., 2024). Additionally, collaboration with C.E.G. enabled the use of energy smart meter data from multiple locations. The data on PV production have been obtained from the PVGIS website (PVGIS, 2024). The selected coordinates were 45.739° N and 7.426° E, corresponding to the Aosta Valley at an elevation of 528 meters, as provided by the website. The solar radiation database employed PVGIS-SARAH2. The mounting type chosen was Fixed, with optimised slope and azimuth values obtained from the website for the location: Slope: 31 degrees (optimum), Azimuth: -20 degrees (optimum). Furthermore, the PV technology considered was crystalline silicon with a 1 kW system, with system losses set at 14.0% as the default value from the interactive tool. The dataset comprises 8,760 hourly values for the years 2005-2020, with an additional 8,784 values for the leap years (2008, 2012, 2016, and 2020), accounting for the extra day. To obtain an annual average, the hourly values were averaged across all years.

The collected one year of real-measured data consists of electricity power curves from three PoDs located in Northern Italy. The dataset includes the following information: sample date, PoD identification, daily-packed sample values of consumption profiles, and energy measurement type. Data preprocessing has been necessary to convert the raw data format into a tabular dataset. The time resolution from 15 minutes has been converted into hourly averages to align with other data sources, such as market energy prices. Data cleaning involved removing duplicates and filtering the PoDs with

the highest number of data points in the studied time frame. The three selected PoDs, shown in Figure 4 have 300 days of real-measured data from June 2023 to June 2024, and represent: a larger consumer, a medium consumer, and a smaller consumer. In cases of missing data due to gaps (less than 1% of the overall dataset), data imputation was performed to create a complete one-year set of energy consumption values. The K-Nearest-Neighbor model has been used as the imputer, one of the standard benchmark methods studied in the literature (Kim et al., 2017).

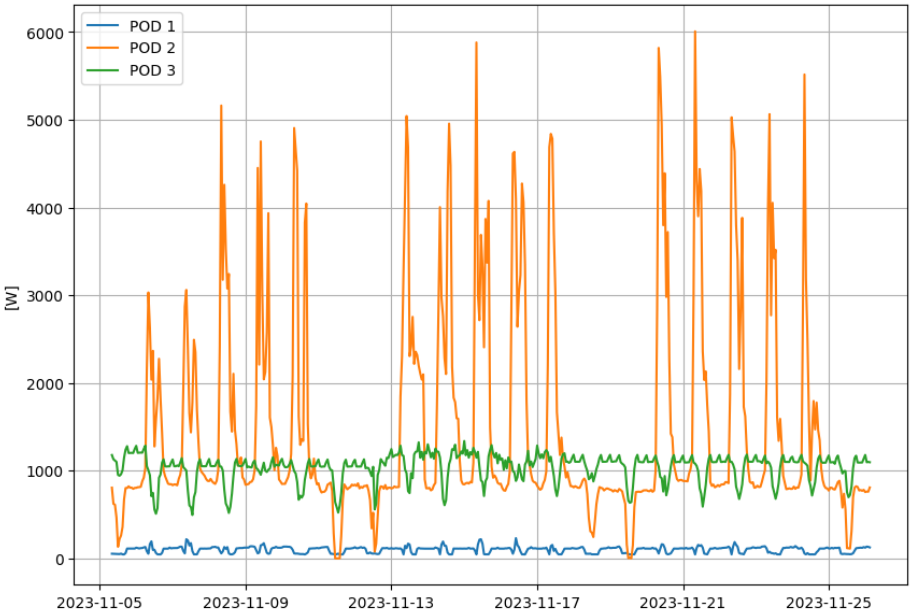


Fig. 4. Exemplary hourly-averaged power profiles.

Table 1. PoD profiles statistics [W].

PoD	min	max	avg	std-dev
1	33	461	96	52
2	0	6477	1067	1005
3	0	2392	1024	334

Since the case study discussed in this paper is located in Italy, the annual incentive is based on the computation of the hourly Italian Premium Tariff (*TIP*) (Gestore dei Servizi Energetici, 2024b) over the course of a year. The financial incentive associated with energy sharing inside the REC is calculated through two main terms, $v_1(t)$ and

$v_2(t)$, in equation 19, which represent different contributions to the total revenue. These terms depend on the energy shared by each PV plant, $\bar{e}_i(t)$, during the period t.

$$I_y = \int_y v_1(t) + v_2(t)\, dt \tag{19}$$

The first term, $v_1(t)$, is calculated using $\mathit{TIP}_i(t)$, as shown in equation 20. For each PoD i, $\mathit{TIP}_i(t)$ reflects a time-dependent incentive parameter that accounts for market prices and other regulatory adjustments. Its calculation is provided in equation 21 and incorporates several factors, including capacity thresholds, market conditions, regional corrections, and governmental contributions.

$$v_1(t) = \sum_{i=1}^{n} \bar{e}_i(t) \times \mathit{TIP}_i(t) \tag{20}$$

$$\mathit{TIP}_i(t) = \{\min\left[\mathit{CAP}_i, \mathit{TP}_i + \max(0, 180 - p^B(t))\right] + \mathit{FC}_R\} \times (1 - F_i) \tag{21}$$

The parameters CAP_i and TP_i depend on the installed PV capacity of the i-th PoD and are divided into three categories summarized in Table 2.

Table 2. TIP table in €/MWh.

$P^{PV}{}_{\text{kW}}$	CAP €/MWh	TP €/MWh
$P < 200$	120	80
$200 \leq P < 600$	110	70
$P \geq 600$	100	60

The parameter CAP_i represents the maximum allowable incentive for the i-th PoD based on its installed PV capacity, as defined in Table 2. TP_i denotes a baseline incentive that is adjusted based on the electricity market price, $p^S(t)$, at time t. If $p^S(t)$ is below 180 €/MWh, an additional incentive proportional to the difference $(180 - p^S(t))$ is added to TP_i. This adjustment ensures that lower market prices do not adversely impact the financial viability of energy sharing within the REC. The term FC_R is a regional correction factor. This factor is set to 10 €/MWh for RECs in Northern regions and 4 €/MWh for those in Central or Southern regions. Finally, the entire expression is scaled by $(1 - F_i)$, where F_i represents the percentage of governmental contribution toward the capital expenditure of the i-th PoD. For simplicity, F_i is assumed to be zero in this analysis, meaning no adjustments are made for governmental contributions.

This formulation ensures that $\mathit{TIP}_i(t)$ dynamically adjusts to market conditions, regional differences, and capacity thresholds, providing a tailored incentive for each PoD in the REC. The energy purchasing price p^B has been set as 282.90 €/MWh, as the average of the gross domestic energy price in 2023 (Autorità di Regolazione per Energia Reti e Ambiente, 2023).

On the other hand, the second term, $v_2(t)$, represents an additional financial incentive associated with energy sharing and is directly linked to the avoidance of transmission losses. It quantifies this benefit, providing a financial acknowledgment of the value created through local energy sharing. This term is calculated using a fixed coefficient, K_{tr}, which applies uniformly to all PoDs, as shown in equation 22. Since K_{tr} is applied uniformly across all PoDs, this term scales proportionally with the total energy shared within the REC, $\bar{e}(t)$, ensuring that larger RECs or those with higher energy-sharing capacities receive greater incentives for avoiding transmission losses.

$$v_2(t) = \sum_{i=1}^{n} \bar{e}_i(t) \times K_{tr} \tag{22}$$

The coefficient K_{tr} is set at a constant rate of 10.57 €/MWh, based on 2023 regulatory values (Gestore dei Servizi Energetici, 2024a). This incentive mechanism is designed to reward RECs for reducing energy transmission distances by sharing energy locally within the community.

No specific requirement for checking whether the investment exceeds a certain budget was included, as the objective was to compare the total costs with and without the REC. This approach was chosen due to the lack of information about the users and their budget. Furthermore, by not setting a maximum acceptable cost, it allowed for the determination of whether the optimal solution provided by the model was the maximum legal capacity of 1MW or an alternative option with lower power levels.

The optimisation has been applied to the specified input data. In this particular case, the selected optimisation engine is Genetic, with the number of iterations set to 10,000. The objective is to find the optimal combination of nominal PV capacity installed on each PoD (ranging from 0 to 1 MW, as per Italian regulations, in increments of 1 kW) to maximise the REC benefits while ensuring continuous electricity coverage for all buildings.

No replacement costs have been factored in, as the lifetime of the PV system is 24 years, while the project's lifespan is considered to be 20 years. d is the discount rate, which has a default value of 7%, while the default percentage value p for OPEX calculation has been assumed equal to 1 [%].

7.1 Results

Three distinct scenarios have been defined, each with a different hourly energy price. Scenario 1 corresponds to 2023, the most recent year for which data is available, with an average hourly energy price of 128 €/MWh over the entire year. Scenario 2 reflects the maximum average hourly energy price over the past six years, which occurred in 2022, at 308 €/MWh. Scenario 3 represents the minimum average hourly energy price over the same period, recorded in 2020, at 38 €/MWh. This approach allows for a comparison under consistent conditions between the most recent year, as well as the highest and lowest price years, to study the benefits of installing a REC under both high and low energy price conditions with the actual incentives regulation.

The hourly energy price data, sourced from the Italian Energy Market Agency (Mercato Elettrico, 2023), correspond to the entire years mentioned earlier and are based on

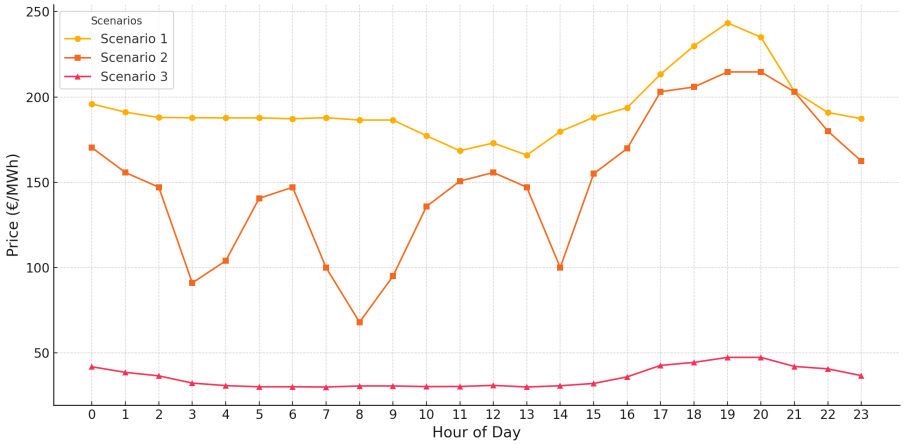


Fig. 5. Hourly Energy Prices in €/MWh for different scenarios.

the North market segment. A sample of this data, representing January 1st, is shown in figure 5.

First, the three scenarios without REC were first executed and the results are provided in Table 3. The analysis confirms that the results are closely aligned across the scenarios since the model has been constructed with identical inputs except for variations in the hourly energy price, which, being the selling price of electricity, does not affect the case without REC. Indeed, no yearly electricity selling is observed in any scenario, as the case without REC does not permit PoDs to sell electricity. Scenario 3 exhibits a slightly higher total cost compared to Scenarios 1 and 2. This difference arises due to 2020 being a leap year, including an additional day (February 29), which marginally increases the yearly buying value and total costs.

Table 3. Comparison of all the scenarios No REC (Yearly).

	Buying [€]	Selling [€]
Scenario 1	5,333	0
Scenario 2	5,333	0
Scenario 3	5,344	0

The Table 4 highlights how variations in average energy prices influence the energy dynamics within the REC. The optimisation results for Scenario 1 converge to a nominal power of 1 kW for Power 1, 7 kW for Power 2, and 4 kW for Power 3. As the average energy price increases, the nominal power of each PoD rises or remains constant across all PoDs. However, the total installed power across the REC increases overall. This increase in installed power results in a reduction in yearly electricity buying, decreasing from 4246 kWh in Scenario 3 to 3765 kWh in Scenario 1 and 3636 kWh in Scenario 2.

Table 4. Comparison of all the scenarios with REC.

Scenario	Nominal Power [kW]			Electricity [kWh]	
	P1	P2	P3	Buying	Shared
Scenario 1	1	7	4	3,765	337
Scenario 2	1	9	5	3,636	284
Scenario 3	0	4	2	4,246	370

Table 5. Comparison of all the scenarios REC Benefits.

	REC Benefit [€]
Scenario 1	7,309
Scenario 2	8,656
Scenario 3	4,179

This aligns with the principle that higher installed power reduces the need to buy energy externally. Conversely, as the average energy price increases, the energy shared within the REC decreases. This is due to greater self-consumption by individual PoDs, leading to less energy available for sharing among them, which in turn reduces the incentives received.

In Table 5 the REC benefits shown. They are highest when the average energy price is maximum, decreasing as energy prices drop, from €8656 in Scenario 2 to €7309 in Scenario 1, and finally to €4179 in Scenario 3. This demonstrates that RECs are more financially advantageous in scenarios where average energy prices are higher.

8 Conclusions and Future Work

This paper presents a multi-method model for simulating and optimising RECs. Developed as part of the PROBONO project, this model represents a significant advancement in the design and management of RECs by enabling detailed performance analysis. Simulation offers detailed insights into electricity demand, production, and energy sharing, alongside evaluating the economic performance of RECs over time. Optimisation identifies the best renewable energy configurations to maximise REC participation benefits while minimising costs.

The model's adaptability and scalability make it suitable for diverse applications, ranging from small to large-scale systems. It allows for adjustments to parameters such as geographical location, PV capacity, and energy prices, ensuring applicability across different regions and regulatory environments. Furthermore, the model's ability to simulate dynamic agent strategies based on national regulations improves decision-making and extends its potential for widespread deployment.

A dedicated tool has been developed to implement the model, allowing users to apply it effectively in diverse contexts and scenarios, bridging the gap between the theoretical framework and practical implementation. Validated through a case study in northern Italy, this tool successfully optimises PV sizes across various energy price

scenarios, showcasing the model's applicability and effectiveness in balancing self-consumption and energy sharing within RECs. The study underscores the critical role of strategic planning in REC configurations to maximise both economic and environmental benefits, with higher energy prices further enhancing the financial advantages of REC participation.

Next steps involve expanding the model's use across the PROBONO Living Labs to assess its adaptability in different contexts. Integration of additional technologies, such as heat pumps, wind turbines, biomass, and hydrogen systems, will advance the model towards multi-sector integration. Geographic factors and advanced energy management systems, including district heating and storage solutions, will further enhance its capabilities. Future studies will also explore the interaction between multiple RECs to optimise energy production and consumption on a larger scale, increasing efficiency and resilience.

Acknowledgment

The authors would like to express their gratitude to C.E.G. Società Cooperativa Elettrica Gignod, Italy, for providing the data used in the real use case. This research was conducted within the framework of The Integrator-centric approach for realising innovative energy-efficient buildings in connected sustainable green neighbourhoods, a project funded by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 101037075. The authors acknowledge that this output reflects only their views, and the European Union cannot be held responsible for any use that may be made of the information contained herein.

References

AnyLogic Company (2024). Anylogic. https://www.anylogic.com/.

Autorità di Regolazione per Energia Reti e Ambiente (2023). Scheda tecnica: L'aggiornamento delle condizioni di tutela, 4° trimestre 2023 nel dettaglio. Available at https://www.arera.it/fileadmin/allegati/schede/230928st.pdf.

Barabino, E., Fioriti, D., Guerrazzi, E., Mariuzzo, I., Poli, D., Raugi, M., Razaei, E., Schito, E., and Thomopulos, D. (2023). Energy communities: A review on trends, energy system modelling, business models, and optimisation objectives. *Sustainable Energy, Grids and Networks*, 36:101187.

Caliano, M., Delfino, F., Somma, M. D., Ferro, G., Graditi, G., Parodi, L., Robba, M., and Rossi, M. (2022). An energy management system for microgrids including costs, exergy, and stress indexes. *Sustainable Energy, Grids and Networks*, 32:100915.

Capros, P., Kannavou, M., Evangelopoulou, S., Petropoulos, A., Siskos, P., Tasios, N., Zazias, G., and DeVita, A. (2018). Outlook of the eu energy system up to 2050: The case of scenarios prepared for european commission's "clean energy for all europeans" package using the primes model. *Energy Strategy Reviews*, 22:255–263.

Cruz-De-Jesús, E., Martínez-Ramos, J. L., and Marano-Marcolini, A. (2023). Optimal scheduling of controllable resources in energy communities: An overview of the optimization approaches. *Energies*, 16(1).

Cutore, E., Volpe, R., Sgroi, R., and Fichera, A. (2023). Energy management and sustainability assessment of renewable energy communities: The italian context. *Energy Conversion and Management*, 278:116713.

Frieden, D., Tuerk, A., Neumann, C., d'Herbemont, S., and Roberts, J. (2020). Collective self-consumption and energy communities: Trends and challenges in the transposition of the eu framework. *COMPILE, Graz, Austria*.

Gestore dei Servizi Energetici (2024a). Allegato 1: Regole operative per i gruppi di autoconsumatori e le comunità di energia rinnovabile.

Gestore dei Servizi Energetici (2024b). Gruppi di autoconsumatori e comunità di energia rinnovabile. Accessed: 2024-12-08.

Gianaroli, F., Preziosi, M., Ricci, M., Sdringola, P., Ancona, M. A., and Melino, F. (2024). Exploring the academic landscape of energy communities in europe: A systematic literature review. *Journal of Cleaner Production*, 451:141932.

Kim, M., Park, S., Lee, J., Joo, Y., and Choi, J. K. (2017). Learning-based adaptive imputation method with knn algorithm for missing power data. *Energies*, 10(10).

Koltunov, M., Pezzutto, S., Bisello, A., Lettner, G., Hiesl, A., van Sark, W., Louwen, A., and Wilczynski, E. (2023). Mapping of energy communities in europe: Status quo and review of existing classifications. *Sustainability*, 15(10).

Lazzari, F., Mor, G., Cipriano, J., Solsona, F., Chemisana, D., and Guericke, D. (2023). Optimizing planning and operation of Renewable Energy Communities with genetic algorithms. *Applied Energy*, 338:120906.

Listopad, S. (2019). Architecture of the hybrid intelligent multi-agent system of heterogeneous thinking for planning of distribution grid restoration. *Baltic Journal of Modern Computing*, 7(4):487–499.

Lode, M., te Boveldt, G., Coosemans, T., and Ramirez Camargo, L. (2022). A transition perspective on Energy Communities: A systematic literature review and research agenda. *Renewable and Sustainable Energy Reviews*, 163:112479.

Mercato Elettrico (2023). Italian Electricity Market Website. https://www.mercatoelettrico.org/It/Default.aspx.

Mihailovs, N. and Cakula, S. (2020). Dynamic system sustainability simulation modelling. *Baltic Journal of Modern Computing*, 8(1):192–201.

OptQuest Website (2024). Optimizing AI/ML hyperparameters with SimWrapper and OptQuest. https://bit.ly/optquest.

Orlando, M., Bottaccioli, L., Quer, S., Poncino, M., Vinco, S., and Patti, E. (2023). A framework for economic and environmental benefit through renewable energy community. *IEEE Systems Journal*, 17(4):5626–5635.

Pagnini, L., Bracco, S., Delfino, F., and de Simón-Martín, M. (2024). Levelized cost of electricity in renewable energy communities: Uncertainty propagation analysis. *Applied Energy*, 366:123278.

PROBONO Project (2022). PROBONO Project Website. https://www.probonoh2020.eu/.

PVGIS (2024). Pvgis official website. https://re.jrc.ec.europa.eu/pvg_tools/en/.

REScoop Website (2024). Transposition tracker: Rec cec definitions.

Sanfilippo, S. and et al. (2023). Microgrid design optimization in benin within the leopard project: Evaluating the impact of inaccurate load profile estimation. In *Proceedings of the 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*.

Sassone, A., Ahmed, S., and D'Angola, A. (2024). A profit optimization model for renewable energy communities based on the distribution of participants. In *2024 IEEE International Conference on Environment and Electrical Engineering and 2024 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pages 1–6.

Sokolowski, M. M. (2018). European law on the energy communities: A long way to a direct legal framework. *European Energy and Environmental Law Review*, 27(2):60–70.

Vetrò, F. and Brignoli, N. (2025). Renewable energy communities: Paradigmatic example of a new decentralised governance of the energy market. *Athens Journal of Law*, 11:25–44.

Zacepins, A., Kviesis, A., Komasilovs, V., and Bumanis, N. (2019). Model for economic comparison of different transportation means in the smart city. *Baltic Journal of Modern Computing*, 7(3):354–363.

Received December 11, 2024 , revised February 20, 2025, accepted February 26, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 221-232
https://doi.org/10.22364/bjmc.2025.13.1.12

Comparison of the PCA and FLD Approaches in Glial Tumors Classification Systems

Miroslav PETROV, Juliana DOCHKOVA-TODOROVA

Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, Veliko Tarnovo, Bulgaria

m.petrov@ts.uni-vt.bg, doskova@ts.uni-vt.bg

ORCID: 0000-0002-5137-6812, ORCID: 0009-0002-2346-3073

Abstract. Computer-aided diagnosis (CAD) systems based on machine learning methods are an important component of medical practice. Principal Component Analysis (PCA) and Fisher Linear Discriminant (FLD) are among the main linear methods for feature extraction and reduction in recognition tasks. In this regard, a comparative analysis of the efficiency of the systems constructed by PCA and FLD algorithms, respectively, is carried out. As a continuation of our previous research, a classification system model for Magnetic Resonance Images (MRIs) of the brain is built using FLD. By analogy, the built CAD system detects the presence of a glial tumor, with a subsequent two-level and three-level gradation according to the degree of malignancy of the tumor. The input images are again not pre-processed and additional wavelet features are used: the normalized energy of the subimages and their non-normalized Shannon entropy. The comparative analysis of the pair of CAD systems is carried out through the quality measures: F1-score and the Matthews Correlation Coefficient. The validation of the obtained results is based on the diagnosis by three independent radiologists.

Keywords: CAD system, PCA, FLD, MRIs, wavelet transform, minimum distance classifier

1. Introduction

The high resolution of the images obtained by the magnetic resonance technology makes them preferrable for medical imaging of the internal organs of the human body. But the vast number of images in medical databases greatly complicates the direct diagnostic activity of radiologists and leads to errors in analysing the specific data. Therefore, the development of computer-aided diagnosis (CAD) systems is an up-to-date task in the field of medical imaging in the relevant diagnostic task.

The main stages of the operation of such systems are:

- pre-processing of the image (noise removal, segmentation, etc.);
- extraction of image characteristics, as well as their eventual reduction;
- training through marked images;
- grouping using a specific classifier.

One of the most important stages in pattern recognition tasks, and therefore in CAD systems as well, is the extraction and reduction of identity features. The transition from the original sample space to a smaller dimensional space is related to the accuracy and

separability of the data. Fisher Linear Discriminant (FLD) and Principal Component Analysis (PCA) are the two main Machine Learning (ML) methods for reducing this dimensionality by linear projection. The goal of PCA is to find the most accurate representation of the data by minimizing the total projection error. The corresponding projection subspace is defined by the eigenvectors of the scattering matrix of the samples. FLD is a classification method indicating the design direction that maximizes the separability of the sample classes and minimizes the internal scatter within the classes, more precisely, it maximizes their ratio (Fukunaga, 1990). The analysis and comparison of their work efficiency with a certain database is a correctly set task. This problem has been mainly addressed in the framework of people identification. Studies whose experimental results show the better performance of FLD recognition algorithms predominate (Belhumeur et al., 1997). However, in (Martinez and Kak, 2001) experiment results are presented showing the superiority of PCA over FLD approaches. This is observed when the classes of the sample contain a small number of representatives or the sample is obtained by uneven selection. In (Eleyan and Demirel, 2006), two face recognition systems based on PCA and FLD algorithms using a neural network for classification are proposed. A comparative analysis of the performance of the proposed systems and the corresponding conventional systems based on the Euclidean classifier is made. In particular, the obtained results confirm the better performance of FLD systems in both classifiers. In (Eleyan, 2008), a comparative analysis is made between the PCA and FLD algorithms in the context of two approaches to solving the face recognition task: the classical approach, considering the whole face, and the method proposed there dividing the face into regions. In particular, the obtained results show that the FLD approach is more effective than the PCA one on large face databases. For example, based on the conducted t-test, within the second approach, this superiority is expressed by more than 16 % . Recently, interest in these problems has been confirmed in (Mostafa and Hossain, 2020), where the performance of PCA, FLD and simple projection approaches for face recognition is investigated. The conducted experiments determine the average efficiency of the three algorithms depending on the dimension of the corresponding projection space. In particular, the better performance of PCA is confirmed when the number of images is small or the sample is uneven with respect to the underlying distribution.

Based on previous research (Petrov, 2023), the task is set to conduct a comparative analysis of the performance of a pair of CAD systems based on PCA and FLD approaches, respectively, for the detection and classification of glial brain tumors.

In (Petrov, 2023), a model is proposed of a two-level CAD system for classifying MRIs of the brain. The classification system uses image descriptors extracted from the PCA projection space. In fulfillment of the given task, a model of an analogous system is built through the FLD approach. The comparative analysis is performed with the same samples of training and test images, keeping the quality measures: F1-score and the Matthews Correlation Coefficient (MCC).

In the rest of the section, some concepts and publications are introduced in order to clarify the content of the sections to come. Timely diagnosis of any tumour entity is extremely important for the outcome of the treatment of the disease. Brain tumours are grouped depending on their origin, place of occurrence, aggressiveness of development, etc. The subject of this work is the task of the classification of glial tumours arising in the auxiliary cells (glia) of the cerebral cortex. Depending on the type of its germ cell, the considered tumours are divided into Astrocytomas, Oligodendrogliomas and

Ependymomas. According to their malignancy and distribution in body tissues, the World Health Organization (WHO) groups them into four classes (Torp et al., 2022):

- Grade I are usually benign and surgically removable tumours;
- Grade II includes astrocytomas, oligodendrogliomas, and oligoastrocytoma (mixed cell type);
- Grade III comprises of anaplastic astrocytomas, anaplastic oligodendrogliomas, and anaplastic oligoastrocytoma;
- Grade IV contains the most aggressive glial tumour called glioblastoma multiforme.

The tumours of the first two classes are defined as low-grade gliomas (LGG) and those from the last two classes are high-grade gliomas (HGG).

There are numerous publications that have proposed various variants of CAD system architectures. In connection with this work, we will give a brief review of two recent overview studies that present sufficiently the engineering methodologies for brain tumour diagnosis.

In (Toufiq et al., 2021), a systematic study of brain tumour classification systems presented in recent years is conducted. The main components of the CAD system are described, as well as the techniques that accompany them. The used classifiers are examined in detail, and a comparative analysis is made between the supervised and unsupervised clustering methods. The review of the brain tumour classification systems in use contains 79 sources. The second publication is (Kaifi, 2023), in which the types of brain tumours and the ways of their detection through imaging methods are presented. An in-depth analysis of the software used in CAD systems is done. A number of brain tumour segmentation and classification methods using the techniques of machine learning and deep learning is reviewed. The corresponding results for the obtained accuracy of these methods are presented, as well as the medical bases used. The literature review contains 127 sources and presents the latest achievements in the field under consideration. In addition, let us note the existence of some CAD systems for the classification of brain tumors, in which the preprocessing step is not performed (Sarhan et al., 2020; Petrov, 2023).

The rest of the document is organized as follows: in the next section, the main steps of Fisher's linear discriminant analysis for extracting the classification features are given; in Section 3, the problem between sample size and data dimensionality for clustering is discussed; the methodology of the proposed system is discussed in the fourth part; and the results of the conducted experiments, their evaluation and the announced comparative analysis are the subject of Section 5. The paper ends with some concluding remarks.

2. Fisher′s Linear Discriminant

We will now briefly discuss FLD as a supervised feature extraction method in the projection space. The considered MRIs can be represented as a one-dimensional vector of its pixels by sequentially connecting the rows (columns) of the $N \times N$ matrix of these pixels, as $x_i = [p_1, p_2, \ldots, p_d]^T$, where $d = N^2$. Then the training sample containing n images from the K class can be written in the form $X = \{x_i, l_i\}_{i=1}^{n}$, where $x_i \in R^d$, and the labels $l_i \in \{1, 2, ..., K\}$. If $X_k = \{x_i \mid 1 \le i \le n_k, l_i = k\}$, then $X = \mathrm{U}_{k=1}^{K} X_k$ and

$n = \sum_{k=1}^{K} n_k$, where $n_k = |X_k|$. To formulate Fisher's criterion (Fukunaga, 1990), it is necessary to define the following two matrices: the within-class scatter matrix –

$$S_w = \sum_{k=1}^{K} \sum_{x \in X_k} (x - \mu_k)(x - \mu_k)^T; \tag{1}$$

and the between-class scatter matrix –

$$S_b = \sum_{k=1}^{K} n_k (\mu_k - \mu)(\mu_k - \mu)^T, \tag{2}$$

where μ_k and μ are respectively the mean values of the k-th grade data and the entire training sample.

Fisher's criterion gives the optimal direction of projecting the original features in a low-dimensional subspace in which the between-class scatter is as large as possible and the within-class scatter is the smallest possible. If the subspace for the FLD is determined by the set of vectors

$$W = [w_1, w_2, ..., w_p] \in R^{d \times p}, \quad p \le \min(d, K-1), \tag{3}$$

then the solution is obtained by maximizing the function

$$J(W) = \frac{\det\left(W^T S_b W\right)}{\det\left(W^T S_w W\right)}. \tag{4}$$

The matrix (3) is constructed from the generalized eigenvectors w by (S_b, S_w), corresponding to the generalized eigenvalues $\lambda = \frac{w^T S_b w}{w^T S_w w}$, i.e.

$$S_b w = \lambda S_w w. \tag{5}$$

In the case where matrix S_w is invertible, equation (5) can be written as a standard equation for finding eigenvectors and eigenvalues of matrix $S_w^{-1} S_b$ –

$$S_w^{-1} S_b w = \lambda w. \tag{6}$$

Under binary classification ($K = 2$), the optimal design direction can be obtained directly from equation (6′) –

$$w = S_w^{-1}(\mu_1 - \mu_2). \tag{6′}$$

The representation of the original data X_k in the space R^p generated by the vectors $\{w_1, ..., w_p\}$, is given by the formula (7) –

$$Y_k = W^T X_k, \quad k = 1, ..., K. \tag{7}$$

Next, each test image x_t needs to be projected in an analogous way into space R^p. The distribution of x_t is based on the Minimum Distance Classifier (MDC) by assigning it the label l_{k^*}, where

$$k^* = \arg\min_k \left\| w^T \left(\mu_k - x_t \right) \right\|_{iR^p}. \quad (8)$$

3. The Small Sample Size Problem

The problem of the small sample size (SSS) is a major challenge when using FLD. If the dimensionality of the original data exceeds their number ($d > n$), then the within-class scatter matrix is singular – $\det(S_w) = 0$. Numerous methods have been developed to overcome the SSS problem, which appears in tasks from various fields, such as face identification, text recognition, bioinformatics, seismology, etc. A detailed overview of these methods can be found, for example, in (Sharma and Paliwal, 2015). Two such methods, which are used in this work, are presented.

3.1. The Moore-Penrose pseudoinverse matrix

A brief description of the Moore-Penrose (MP) pseudo-inverse for the matrix $S_w \in R_1^{d \times d}$. is given in (Wu, 2017). For this purpose, the spectral decomposition (diagonalization) $S_w = U \Lambda U^T$ is used, where Λ is a diagonal matrix containing the eigenvalues of S_w, and the columns of the orthogonal matrix U contain their respective eigenvectors. Let $\Lambda = diag\left(\left[\lambda_1, \lambda_2, ..., \lambda_d \right]^T \right)$, then its MP pseudo-inverse is set as $\Lambda^+ = diag\left(\left[\lambda_1^+, \lambda_2^+, ..., \lambda_d^+ \right]^T \right)$, where

$$\lambda_i^+ = \begin{cases} 0 & if\ \lambda_i = 0 \\ \lambda_i^{-1} & otherwise \end{cases}, \quad (9)$$

and the MP pseudo-inverse of S_w is $S_w^+ = U \Lambda^+ U^T$. It should be noted that, if S_w is not singular, then $S_w^+ = S_w^{-1}$.

3.2. Robust FLD Model

In (Deng et al., 2006), the Robust Fisher Linear Discriminant Analysis (RFLDA) method is proposed for the case of a singular (or close to singular) matrix S_w. Again, the spectral decomposition of the within-class scattering matrix is considered, and its eigenvalues are sorted – $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$, as well as their corresponding eigenvectors. In this case, the last eigenvalues of the matrix can compromise the results of the discriminant analysis, so in RFLDA they are replaced by a certain unified value. Based on statistical analysis, the number d^* of the main eigenvalues is determined, and the remaining $\left(d - d^* \right)$ are either very small or equal to zero. After conducting experiments, the authors propose the determination of d^* by minimizing the function

$$E\left(d^*\right)=\left(\sum_{i=1}^{d^*}\lambda_i\right)g\left(\sum_{j=1}^{d}\lambda_j\right)^{-1}, \tag{10}$$

so that its values remain within the interval $(0.9, 0.99)$. The remaining $\left(d-d^*\right)$ eigenvalues are then replaced by $\lambda^*=\frac{1}{d-d^*}\sum_{j=d^*+1}^{d}\lambda_j$. Thus, the matrix S_w is evaluated by

$$S_w^*=U\Lambda^*U^T, \tag{11}$$

where $\Lambda^*=diag\left(\left[\lambda_1,...,\lambda_{d^*},\lambda^*,...,\lambda^*\right]^T\right)$, and the design directions are set by the generalized eigenvectors of $\left(S_b, S_w^*\right)$.

4. Methodology

In this part, the methodology of the proposed system is explained. Its design is presented in Fig.1. At first, the brain MRIs selected in the training sample are divided into normal and abnormal, according to the absence or presence of a glial tumour. In the next step, the abnormal images are grouped into two or three classes, depending on the extent of the glial tumour present.

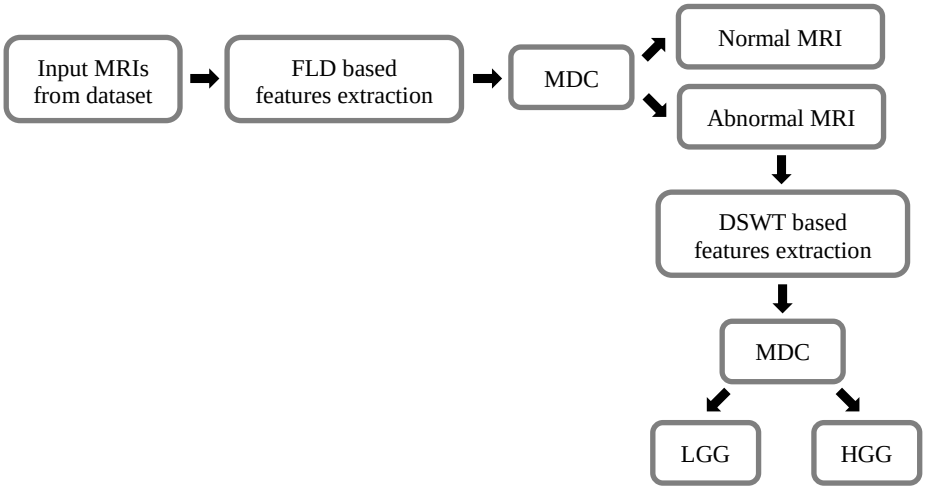


Figure 1. Block diagram of the proposed system.

4.1. Tumour Detection Stage

Let the training sample of MRIs that enter the input of the CAD system be $X = \bigcup_{k=1}^{2} X_k$, where $X_k = \{x_i \mid 1 \le i \le n_k, l_i = k\}$ and $n_k = |X_k|$. The required image features are extracted using FLD. Projecting the original data X into space R is performed according to equation (7). For the classification stage, it is necessary to determine the centroids $w^T \mu_k$ of each of the two classes X_k, $k = 1, 2$. Next, the image received for classification x_t is projected in an analogous way in R and its belonging to one of the two classes is determined by MDC and the corresponding weighted metric.

4.2. Tumour Classification Stage

Let X_a be the array of MRIs with malignant entities obtained at the first stage of the operation of the CAD system. Then the training sample for the second classifications will have the type $X_a = \bigcup_{k=1}^{K} X_{a_k}$, where $K = 2 \vee K = 3$ and $X_{a_k} = \{x_{a_i} \mid 1 \le i \le n_k, l_i = k\}$, $n_k = |X_{a_k}|$, $k = 1, ..., K$. In addition to FLD, the Discrete Stationary Wavelet Transform (DSWT) is also used to extract the required image features (Mallat, 1998). The detailed wavelet coefficients are indicators of the local peculiarities of the signals, therefore only the three sub-bands LH, HL and HH are considered in this work (see Fig.2).

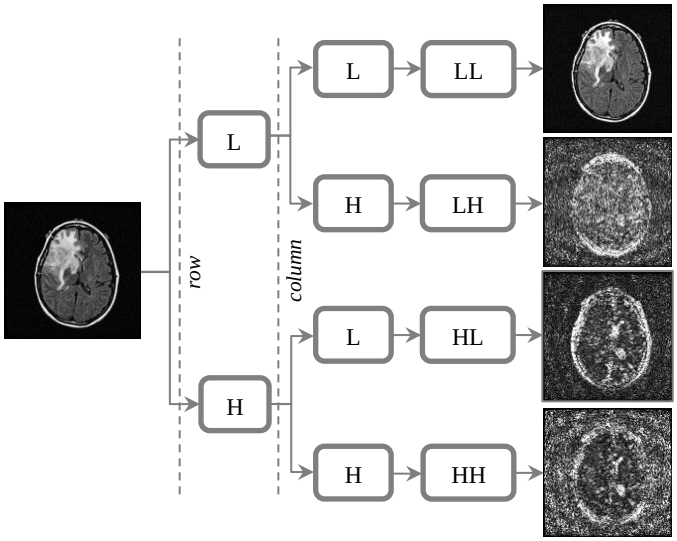


Figure 2. MRI decomposition by DSWT.

The extracted energy and entropy features are added to those obtained by FLD to construct the image descriptor vector. For the second part of this vector, the normalized energy of the sub-image P_{r_s} – $E_{r_s} = d^{-2}\sum_{q,m} D_{r_s}^2(q,m)$, is used, where the corresponding wavelet coefficients are indicated by D_{r_s}. The resulting energy characteristics are $\{E_{r_s}\}$, $r = 1,2,3$ and $s = 1,...,L$, where L is the maximum decomposition level. The third part of the descriptor contains the entropy features $\{H_{r_s}\}$, $r = 1,2,3$, $s = 1,...,L$, where $H_{r_s} = -\sum_{q,m} D_{r_s}^2(q,m)\cdot \log D_{r_s}^2(q,m)$ is the unnormalized Shannon entropy of the subimage P_{r_s}.

Similarly, the feature vector of a random test image x_t, is obtained, and its belonging to the corresponding class is again determined by MDC. The metric used in formula (8) is the Normalized Euclidean Distance (NED).

5. Comparative Analysis and Discussions

In the proposed system, supervised learning is carried out, allowing to take into account the possible errors of the classifier. The performance of the CAD system is evaluated by the measures F1-score and MCC, which are predetermined by the goals set in this work. The first one is the harmonic mean of precision and recall, proportional to the quality of the classifier. The second measure reflects the relationship between the observed and predicted data using the entire confusion matrix and is unaffected by the dimensionality of the classes. Besides, it should be noted that, when conducting the comparative analysis of the performance of the pair of CAD systems, the values of the quality measures were obtained using the expert opinion of three radiologists.

5.1. Accuracy in Tumour Detection

The presented results needed to perform the requested comparative analysis were obtained with the same collection of 340 brain MRIs (Petrov, 2023), 200 of which represent the training sample. T1-weighted (T1W), T2-weighted (T2W) and T2-sensitive (T2F) images were used, each of 256×256 pixels in size in DICOM format. The training samples were labelled by three independent experts, the sample itself being balanced. The data were obtained from the following publicly available medical databases (Pedano et al., 2016; Scarpace et al., 2019; Erickson et al., 2017) and from the Imaging Department of Dr Stefan Cherkezov Hospital of Veliko Tarnovo. Fig. 3 shows MRIs containing glial tumours of the following types: astrocytoma, oligodendroglioma and gliobastoma.

When using the F1-score and MCC measures to evaluate binary classifications and their corresponding confusion matrices, their sensitivity to the balance of the dataset should be considered. MCC uses all elements of the confusion matrix, making it robust to unbalanced samples.

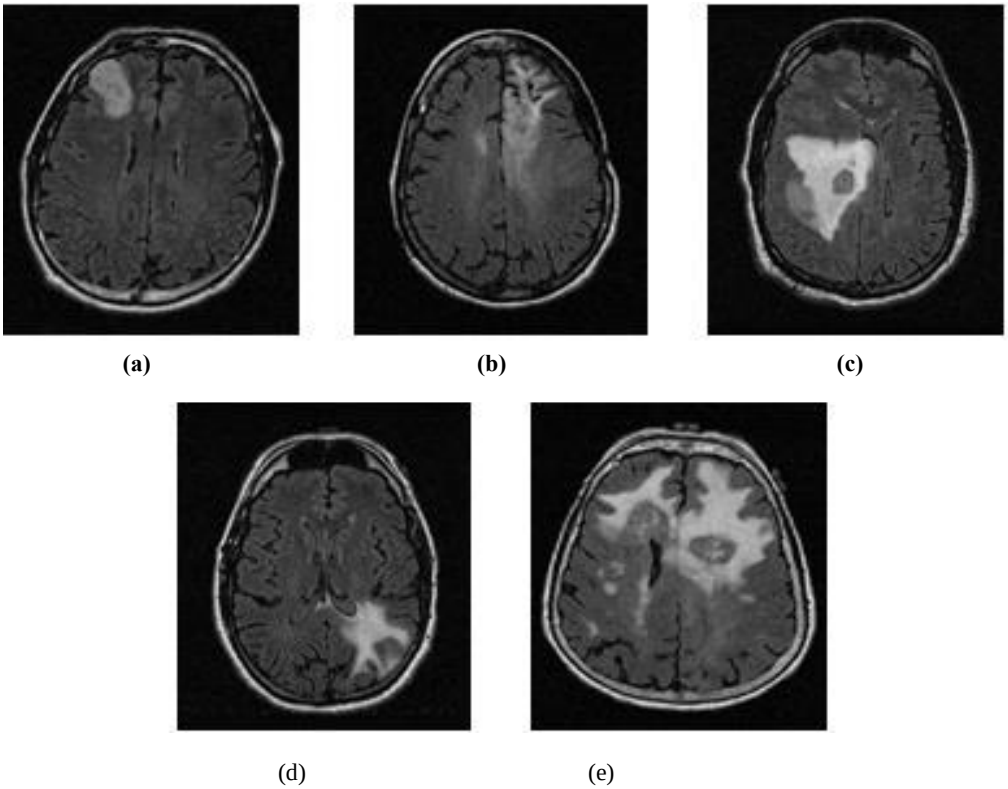


Figure 3. Glial tumors: a) low-grade astrocytoma; b) low-grade oligodendroglioma; c) high-grade astrocytoma; d) high-grade oligodendroglioma; e) gliobastoma.

The results of the comparative analysis in the tumour detection stage are presented in Table 1. Its data show that the FLD methods outperform the PCA method by an average of 3.23% and 11% , respectively, for F1-score and MCC. Furthermore, the RFLDA method shows better classification than the FLD–MP method in both evaluation metrics.

Table 1. Values of the performance measures of the classifier for tumour detection

Methods	**Performance metrics**	
	MCC	**F1-score**
PCA (Petrov, 2023)	0.82	0.93
Proposed FLD - MP	0.89	0.95
Proposed RFLDA	0.93	0.97

5.2. Accuracy in Tumor Classification

In this part, the performance of the CAD system in categorizing the glial tumour grade is investigated. The training sample is obtained from the original one, keeping the malignant tumour MRIs. The test set contains 100 slices of low-grade and high-grade gliomas.

The results of the comparative analysis for the second classification are presented in Tables 2 and 3. The data show that for these classifiers too, the FLD methods perform better, on average by 3.8% (F1-score) and by 3.2% (MCC). Again, the better performance of the second method is confirmed. The data from the additionally performed three-level classification show the superiority of the RFLDA method over the FLD-MP method, with

Table 2. Indicators in the two-stage classification of the tumor

Methods	Performance metrics	
	MCC	F1-score
PCA (Petrov, 2023)	0.78	0.92
Proposed FLD - MP - MP	0.8	0.95
Proposed RFLDA	0.81	0.96

Table 3. Indicators in the three-stage classification of the tumor

Methods	Accuracy in tumour grading [%]		
	II grade	III grade	IV grade
Proposed FLD - MP	88	79	91
Proposed RFLDA	91	85	94

the corresponding percentage expression being 3.4% (for II grade), 7.6% (for III grade) and 3.3% (for IV grade).

5.3. Discussions

The objective set in Section 1 and the descriptors used justify the comparative analysis between the PCA and FLD methods in the tasks of glial tumours detection and clustering. These are two projection methods of ML to reduce the dimensionality of the original data space. PCA maximizes the accuracy of the samples in the projection space while preserving the variance of the original data. FLD is a supervised classification method that tries as much as possible to preserve the necessary information to separate the classes.

The basis for the comparative analysis is that in the present work both methods are used to determine the centroids of the classes. From the data presented in the above tables, it can be seen that the classifiers based on FLD methods are more efficient than those using PCA. The values in each of these tables are obtained by averaging the results of twenty tests with data randomly generated from the respective test samples. From a computational

point of view, the mathematical implementation of the FLD algorithm requires a significant amount of RAM, even for images with a resolution of 256×256 pixels. An additional difficulty is the singularity of the scattering matrix S_w due to the SSS problem.

6. Concluding Remarks

Because FLD is a direct class separation method and PCA is a method representing data as a whole, the former is usually assumed to be superior in recognition tasks. This hypothesis is also confirmed by the comparative analysis carried out in the previous section. But as it was stated in Section 1 there are cases when PCA outperforms FLD in some tasks. For example, if there is a small-size training sample (unrepresentative) or if it is unevenly distributed across classes (unbalanced). The proposed analysis can be extended by examining the performance of the methods: as a function of the sample size; at different class distributions or at additional wavelet features obtained with other multiscale transformations.

References

Belhumeur, P., Hespanha, J., Kriegman, D. Using discriminant eigenfeatures for image retrieval. PAMI, **19**(7), 711–720, 1997.

Deng, W., Hu, J., Guo, J. (2006). Robust Fisher Linear Discriminant Model for Dimensionality Reduction, *18th International Conference on Pattern Recognition (ICPR'06)* **4,** 699-702, doi: 10.1109/ICPR.2006.211.

Eleyan, A., Demirel, H. (2006). PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds) Multimedia Content Representation, Classification and Security. MRCS 2006. Lecture Notes in Computer Science, **4105**. Springer, Berlin, Heidelberg, https://doi.org/10.1007/11848035_28.

Eleyan, M. (2008). PCA and LDA based face recognition using region-division with majority voting. http://library.neu.edu.tr/Neutez/4713191966.pdf.

Erickson, B., Akkus, Z., Sedlar, J., Korfiatis, P. (2017). Data from LGG-1p19qDeletion(version 2) [data set]. The cancer imaging archive, https://doi.org/10.7937/K9/TCIA.2017.DWEHTZ9V.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, California, USA.

Kaifi R. (2023). A Review of Recent Advances in Brain Tumor Diagnosis Based on AI-Based Classification. *Diagnostics (Basel)* **13** (18):3007, 1-32, doi: 10.3390/diagnostics13183007.

Mallat, S. (1998). *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, California, USA.

Martinez, A. M., Kak, A. C. (2001). PCA versus LDA, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23** (2), 228-233, doi: 10.1109/34.908974.

Mostafa, F. B., Hossain, S. (2020). Revisiting the performance of PCA versus FDA versus Simple Projection for Image Recognition. *Global Journal of Engineering and Technology Advances* **8** (1), 084-095.

Pedano, N., Flanders, A. E., Scarpace, L., Mikkelsen, T., Eschbacher, J. M., Hermes et al., (2016). The cancer Genome Atlas Low Grade Glioma Collection (TCGA-LGG) (version 3) [data set]. The cancer imaging archive, https://doi.org/10.7937/K9/TCIA.2016.L4LTD3TK.

Petrov, M., (2023). A model of a two-stage classification system for glial tumors in magnetic resonance imaging, *2023 International Conference Automatics and Informatics (ICAI)*, 418 - 422, doi: 10.1109/ICAI58806.2023.10339009. ISBN 979-8-3503-1291-1.

Sarhan, A. M. (2020). Brain Tumor Classification in Magnetic Resonance Images Using Deep Learning and Wavelet Transform, *J. Biomedical Science and Engineering*, **13** (6), 102-112.

Scarpace, L., Flanders, A. E., Jain, R., Mikkelsen, T., Andrews, D. W. (2019). Data from REMBRANDT[dataset]. The cancer imaging archive, https://doi.org/10.7937/K9/TCIA.2015.588OZUZB.

Sharma, A., Paliwal, K. (2015). Linear discriminant analysis for the small sample size problem: an overview. *Int. J. Mach. Learn. & Cyber* **6**, 443–454, https://doi.org/10.1007/s13042-013-0226-9.

Torp, S. H., Solheim, O., Skjulsvik, A. J. (2022). The WHO 2021 Classification of Central Nervous System tumours: a practical update on what neurosurgeons need to know—a minireview, Acta Neurochir 164, 2453–2464, https://doi.org/10.1007/s00701-022-05301-y.

Toufiq, D. M., Ali Makki Sagheer A. M., Veisi H. (2021). A Review on Brain Tumor Classification in MRI Images, Turkish Journal of Computer and Mathematics Education 12 (14), 1958- 1969.

Wu, H. (2017). Fisher's linear discriminant. https://cs.nju.edu.cn/wujx/paper/FLD.pdf.

Received August 13, 2024, revised January 20, 2025, accepted January 23, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 233-251
https://doi.org/10.22364/bjmc.2025.13.1.13

Applying Word Embeddings for Lithuanian Morphology: The Case of Adjectival Participles

Laima JANCAITĖ-SKARBALĖ, Erika RIMKUTĖ,
Justina MANDRAVICKAITĖ

Vytautas Magnus University, Kaunas, Lithuania

laima.jancaite-skarbale@vdu.lt, erika.rimkute@vdu.lt,
justina.mandravickaite@vdu.lt

ORCID 0009-0005-9210-672X, ORCID 0000-0003-0858-8593, ORCID 0000-0001-9426-6165

Abstract. This paper presents how word embeddings were used to identify adjectival Lithuanian participles. Although traditionally considered to be a form of a verb, participles in the Lithuanian language also have the characteristics of adjectives. The paper describes a study on how one of the criteria for the identification of adjectival participles was applied using the *fastText* word embedding model. This criterion involves the recognition of adjectives and pronouns that are semantically similar to participles (e.g., these adjectives and pronouns can be synonyms or antonyms of participles). The paper assesses the extent to which word embeddings can help to identify adjectival Lithuanian participles and summarises the advantages and disadvantages of this method. Out of 289 analysed participles, 48 participles (16.61%) that are semantically similar to adjectives and pronouns were identified using word embeddings.

Keywords: Lithuanian language, word embeddings, fastText, grammar, adjectival participles

1. Introduction

Lithuanian is a morphologically complex, inflected language (Ambrazas, 2006, p. 89). The most complex and distinctive part of speech are verbs, as some forms are declined (participles, e.g., ***einantis*** *žmogus* “a person **walking**”, ***keptas*** *pyragas* “a **baked** pie”), other forms are conjugated (moods and tenses, e.g., *žmogus* ***eina*** “a man **is walking**”, *mama* ***keptų*** *pyragą* “mother **would bake** a pie”), some forms are minimally inflected (half participles and gerunds, e.g., ***eidamas*** *į universitetą nukritau* “I fell **while going** to the university”), and others are not inflected at all (infinitives, verbal adverbs, e.g., *eiti* “to go”, *kepti* “to bake”; ***bėgte*** *bėgo į universitetą* “ran **in a hurry** to the university”).

Participles have many similarities to adjectives: they have categories of gender, number, and cases, can have degrees, a neuter gender, and often can have an attributive function. On the other hand, they retain the most important categories of verbs: transitivity,

aspect, and reflexivity, and they have similar valency to verbs. Thus, participles have the properties of both adjectives and verbs.

However, some participles are similar to adjectives not only in syntactic functions and grammatical categories but also semantically. In this paper, these participles are referred to as adjectival participles. They may even lose verbal properties - they may change their lexical meaning from the verb and lose the grammatical categories of participles that are not characteristic of adjectives (voice, tense)[1], e.g., *patyręs asmuo* "an experienced person" – *patirti* "to experience"; *papildomas* "additional" – *papildyti* "to add"; *prieinamos kainos* "affordable prices" – *prieiti* "to approach"; *teigiamas dalykas* "a positive thing" – *teigti* "to state"; *niekuo dėtas* "innocent" – *dėti* "to put". It is essential to identify adjectival participles as they not only differ from other participles in meaning but also can begin to function as separate lexemes compared to other verb forms and form certain exceptions to usage. Since these participles begin to function as a separate lexeme, they can have separate lemmas from the corresponding verbs in dictionaries; identifying them is also vital in teaching Lithuanian as a foreign language; moreover, they are important in the morphological and syntactic annotation of corpora, in conducting various linguistic studies, and in the writing of grammars.

A study is being carried out, aiming to identify adjectival participles in an objective and partially automated way. It attempts to identify them based on various criteria: grammatical criteria (frequent attributive function; no verbal arguments and words indicating place and time; combination with adverbs of measure/degree; gradation (*mylimas* "beloved" – *mylimiausias* "most beloved"); frequent pronominal (definite) form; semantic criteria (change in lexical meaning from the corresponding verb; synonyms and antonyms with adjectives, the same semantic classes with adjectives); derivational criteria (tendency to form adverbs with the suffix *-ai* and abstract nouns with the suffixes *-umas* and *-ybė* from participles); and quantitative criteria (frequent use of participles in corpora compared to other forms of verbs). The criteria of adjectival participles are also mentioned in Jancaitė (2023).

Some of the mentioned criteria for adjectival participles (especially the semantic criteria) appear to be very subjective, time-consuming, so it was important to find a way to apply these criteria in a more objective manner (using corpus data). The aim of this paper is to present how one semantic criterion for adjectival participles (participles that have synonyms and antonyms similar to adjectives and share the same semantic classes as adjectives) was applied in a more objective way, based on word embeddings methodology.

The analysis was not limited to adjectives, as it is assumed that some participles can be used like pronouns (e.g., *praeitą* / *aną savaitę* "last week": *praeitą* is a participle, *aną* is a pronoun). They are also different from other participles, so it was decided to analyse which participles are similar to adjectives and pronouns.

[1] Participles that lose their verbal properties but gain adjectival properties can be called adjectivised participles (cf. Petrunina, 2021, Paulauskienė, 1994, Kustova, 2021). However, some participles that are semantically similar to adjectives still retain the meaning of a verb (e.g., *miręs* "dead" - *mirti* "to die"). In this paper, participles with adjectival meaning (whether they have lost verbal properties or are related to verbs) are referred to as adjectival participles.

The sources of the study are Pedagogic Corpus of Lithuanian[2] (hereinafter referred to as Pedagogic Corpus), as well as in the Lexical Database of Lithuanian Language Usage[3] (hereinafter referred to as Lexical Database) and pre-trained *fastText* model, freely available on the internet[4]. The Pedagogic Corpus is a corpus designed for learners of Lithuanian at different levels of proficiency; it contains 699,000 words, comprising written (619,000 words) and spoken (35,000 words) texts. For more details on the Lexical Database, see Kovalevskaitė et al. (2022). The *fastText* model was trained on a corpus with social media comments (comments taken from Delfi.lt and lrytas.lt, written between 18-09-2014 and 30-05-2020, totalling 2 million comments). This corpus consists of 58,042,082 words, including 2,306,198 unique words.

Chapter 2 describes the word embedding approach that was crucial for identifying adjectival participles. Chapter 3 outlines the research methodology and presents the research results. The conclusions are then provided, followed by appendices containing analysed participles with and without semantically similar adjectives or pronouns.

2. Word Embeddings in Linguistic Analysis

Word embeddings show semantic relationships between words based on distributional (syntactic and lexical) information. Using word embeddings, words can be grouped by distribution into groups – word classes (see Kutuzov et al., 2016), word synonyms can be identified, and other studies can be carried out (cf. Gutiérrez and Keith, 2019).

Word embeddings are related to the research area of distributional semantics, which was founded on the ideas of Firth (1962), Harris (1954), and other researchers. The distributional hypothesis was derived, which states that words that are used and occur in the same contexts tend to have similar meanings (Firth, 1962). For instance, Harris (1954, p. 156) noticed that synonyms (like *oculist* and *eye-doctor*) tend to occur in the same environments. However, words that are not semantically similar will not share the same environments; for instance, *oculist* and *eye-doctor* may occur near words like *eye*, *examined*, but *lawyer* probably will not.

Thus, the idea that the meaning of a word is determined by context was already discussed in the first half of the 20th century. As technology has advanced, it has become possible to automatically analyse large volumes of texts, allowing for the automatic analysis of word distribution on a large scale.

Vector semantics is important to automatically estimate which words are used in similar contexts. In vector semantics, a word is represented as a point in a multidimensional semantic space that is derived from the distributions of neighbouring words. Vectors that represent words are called embeddings (Jurafsky and Martin, 2024, p. 105). These embeddings encode the characteristics of the words and the contexts in which they are used. This is also how similar words can be identified – similar words are represented by embeddings in a nearby space. Thus, words belonging to the same semantic classes, such

[2] https://kalbu.vdu.lt/en/resources/pedagogic-corpus-of-lithuanian/
[3] https://kalbu.vdu.lt/mokymosi-priemones/leksikonas/
[4] http://fasttext.vdu.lt/

as synonyms, antonyms, hyponyms, hyperonyms, and other related words can be identified (this is also reported in Kovalevskaitė et al., 2021; in this study, arbitrary collocations were identified using word embeddings).

Word embeddings are used for a diverse variety of tasks and cases, e.g., for domain-specific representations (Wang et al., 2021, Brandl et al., 2022), to discriminate inflection and derivation (Haley et al., 2023), detect semantic shifts in inflectional morphology (Gromann and Declerck, 2019), produce large analogical clusters (Hong and Lepage, 2018), distinguish denominal and root-derived verbs (Benbaji et al., 2022), predict semantic priming (Kastner, 2020), and evaluate gender bias in word representations (Altinok, 2024), to name just a few.

However, word embeddings have been used only to a limited extent in research of the Lithuanian language: examples include using *GloVe* embeddings for analysing arbitrary collocations (Kovalevskaitė et al., 2021) and identifying Lithuanian multiword expressions (Bumbulienė et al., 2018), developing a recogniser of hate/offensive speech in social media texts (hatespeech.vdu.lt). Word embeddings have also been applied in sentiment analysis of Lithuanian texts (Kapočiūtė-Dzikienė et al., 2019). Different types of word embeddings for Lithuanian have been used in deep learning cases such as news clustering (Stankevičius and Lukoševičius, 2020), automatic extraction of cybersecurity terms (Rokas et al., 2020), and the identification of semantic and syntactic similarity/relatedness (Petkevičius and Vitkutė-Adžgauskienė, 2021).

3. Word Embeddings for Identifying Adjectival Participles

Since one of the criteria for identifying adjectival participles is their semantic similarity with adjectives (including their synonymy, antonymy, and tendency to share the same semantic class), word embeddings were tested to automatically identify participles that are distributionally similar to adjectives and pronouns. Although distributional similarity is not the same as semantic similarity, it is believed to help detect semantic similarity in a more objective way than relying solely on introspection.

3.1. Methodology

As mentioned in the introduction, the sources of the study are the Pedagogic Corpus, the Lexical Database, and the pre-trained *fastText* model. The study includes 289 participle lemmas[5]. Lemmas were selected from the 200 most frequent verbs in the Lexical Database, e.g., *mylėti* “to love”; *žinoti* “to know”; *eiti* “to go”. Only participles with at least 5 occurrences in the Pedagogic Corpus in the simple (indefinite) form[6] were

[5] Usually, verb lemma also covers participles, e.g., *dirbti* “to work” also covers *dirbantis žmogus* “working person”. In this paper, nominative participle forms like *dirbantis* are called participle lemmas.

[6] The pronominal (definite) forms of adjectives, participles, etc., indicate a specific, known object (e.g., *gražusis* “the beautiful one”), while the simple forms do not mark definiteness but only

analysed; in this study, pronominal (definite) participles were not included (e. g., *valgomas* "edible, eaten" was included, but not "valgomasis").

Initially, the idea of the research was to use a word embedding model trained with Pedagogic Corpus data (*fastText* and *word2vec* models were trained), as this corpus was used to select analysed words and analyse them. However, it turned out that the models trained with Pedagogic Corpus data had too little training data, resulting in a lack of accuracy (since this corpus is relatively small). In the end, the pre-trained *fastText* model, freely available on the internet[7] was chosen.

FastText represents words as bags of character n-grams to capture sub-word information, making it robust to out-of-vocabulary words. *FastText* is based on *word2vec* as it uses skip-gram (it predicts context words when given a target word) and Continuous Bag of Words (CBOW; it predicts a target word when given context words) methods (Bojanowski et al., 2016; Naseem et al., 2021). It became popular due to being fast and efficient and due to models being available for a large number of languages (Joulin et al., 2017). Since *fastText* models do not ignore the morphology of words, i.e., they take into account the internal structure of words, the information about linguistic units smaller than the word, they are said to be more suitable for morphologically rich languages such as Lithuanian (other techniques associate each word with a distinct vector without parameter sharing (Bojanowski et al., 2017). However, since this model ranks words with similar morphology as more similar, it was decided in this research to search for the most frequent forms of words rather than their lemmas. Therefore, distributionally similar words for each word form were identified first (see Table 1). Later, the participles and the identified similar words were assigned to a single lemma. For each word form, 50 distributionally similar words were found. During this research, when comparing words, a certain estimate of word similarity was obtained – the closer it is to 1, the more similar the word is (see Table 1).

When analysing the participles, first of all, the 3 most frequent forms of the participles from the Pedagogic Corpus were entered into the search (for an example of a search with similar words, see Table 1), for instance, the forms of the participle *verdantis* "boiling": *verdantį, verdančio, verdančiu*. If a participle in the Pedagogic Corpus has a comparative or superlative degree, the most frequent forms of the comparative/superlative degrees were also searched for separately (e.g., in addition to some forms of the participle *lankomas* "visited", the superlative form *lankomiausias* "the most visited" was also searched for). However, if no distributionally similar adjectives or pronouns were found among these most frequent forms, all forms of the participles used in the Pedagogic Corpus were searched.

Thus, the initial search focused on adjectives and pronouns (see Table 1).

---

describe the characteristic of an object, without indicating definiteness (e.g., *gražus* "beautiful"). In Lithuanian, definite adjectives are formed by adding a suffix such as *-is* (masculine), *-oji* (feminine), etc. (Valeckienė, n.d.).

[7] https://fasttext.vdu.lt/. During the pilot study, another *fastText* model—a Lithuanian model trained on a corpus from *Wikipedia* and Common Crawl, available as pre-trained word vectors from the *fastText* project (Grave et al., 2018)—was applied. However, due to uncertainty regarding the number of trained tokens and the nature of the texts in Lithuanian, this model was later disregarded.

Table 1. Similar words analysis using word embeddings.
A search for the participle *praėjusį* ("past", "passed") (highest score is 1)

Nr.	Word	Score
1	Praėjusį	0.89438444
2	praėjusią	0.88527215
3	patį.Praėjusį	0.8692309
4	praeitą	0.8532955
5	"Praėjusį	0.8373966
6	užpraeitą	0.8287578
7	praeitą,	0.8278644
8	pirmąjį	0.81250143
9	Praėjusią	0.81066614
10	Įvykusį	0.805664
11	prasidėjusį	0.80437165
12	įvyksiantį	0.7967169
13	praeitą.	0.79365396
14	neįvykusį	0.7915372
15	būsiantį	0.79064226
16	praitą	0.7906226
17	pastarąjį	0.7896867
18	vykusį	0.78891814
19	besitęsiantį	0.78527635
20	šiųmetinį	0.78449905
21	"Praėjusią	0.7789983
22	užsitęsusį	0.7773801
23	(paskutinį	0.77316064

As can be seen in Table 1, there was also a lot of noise in the analysis – the results include different forms of the same participle (e.g., *praėjusią, Praėjusią*) and the participles were very often recognised as similar to other participles (e.g., *užpraeitą* "last"/"previous", *įvykusį* "occured"*, prasidėjusį* "started", *įvyksiantį* "that will happen"). Also, letters were not converted to lower case, and punctuation marks were not separated from words. Thus, at this stage of the research, linguistic editing was necessary, i.e.,

adjectives and pronouns were selected from all the similar words found. In this case (see Table 1), the adjectives selected are the following: *pastarąjį* "last", *šiųmetinį* "this year's", *paskutinį* "last".

Once the adjectives and pronouns were selected, it was important to check whether they are really semantically similar to the participle under analysis (As mentioned earlier, although word embeddings can help identify semantically similar words, this is not always the case, because word embeddings are based on statistics – they reflect distributional, rather than semantic, similarity, and this does not always align). For the purposes of this study, semantically similar words are assumed to be those that are used synonymously and antonymously and belong to the same semantic class. For instance, in this case (see Table 1), it has been recognised that the participle *praėjęs* "last" is used synonymously with the adjective *pastarasis* "last" and that this participle with the adjectives *pastarasis* "last" and *šiųmetinis* "of this year" belong to the same semantic class denoting time (for example, *praėjusią / pastarąją savaitę* "last week"). In another example, the participle *verdantis* "boiling" is synonymous to the adjective *karštas* "hot", both of which belong to the semantic class denoting temperature. As mentioned before, the analysis was not limited to adjectives, as it is assumed that some participles can be used like pronouns, e.g., *praeitą* / *aną savaitę* "last week" (*praeitą* is a participle, *aną* is a pronoun).

Linguistic editing was also important at this stage of the study, as it was observed that the selection of the adjectives and pronouns resulted in some not being semantically similar to the participles (e.g., *nepamirštamas* "unforgettable" – *simboliškas* "symbolic"; *papildomas* "additional" – *periodinis* "periodic"). It is likely that these participles and adjectives were identified as similar because they are used in similar contexts, e.g., in similar terms. As mentioned before, distributional similarity does not always reflect semantic similarity.

However, without the context of usage, it is sometimes difficult to determine whether words are semantically similar or not. Therefore, during the linguistic editing stage, it was important to check the context of usage as well. The difficulty is that using this method, the context of words could not be seen, only similar words (as shown in Table 1). Therefore, the Pedagogic Corpus was used for this task. Moreover, participles can have multiple meanings. When analysing word embeddings data, it is not always clear which meaning is being represented by the similar words, as the wider context is not visible. For this reason, the concordances of the analysed participles in the Pedagogic Corpus were checked to determine the semantic context in which the participles are semantically similar to the adjectives or pronouns detected by the word embeddings (and whether such a context really exists). For example, the participle *praėjęs* (see Table 1) is semantically similar to the adjectives *pastarasis* and *šiųmetinis* in the following case:

*Čempionate neliko **praėjusį** sezoną paskutinės vietos užėmusios Elektrėnų „ESSM-2000" komandos.*

"**Last** season's last-place team, Elektrėnai ESSM-2000 did not participate in the championship. "

In this case, the participle *praėjusį* can be replaced by adjectives *pastarąjį* "last" and *šiųmetinį* "this year's".

However, there are cases where the participle *praėjęs* cannot be considered semantically similar to the adjectives *pastarasis* and *šiųmetinis* because it conveys a

different meaning of the participle – not "last", but "after passing" (this participle is semantically similar to the verb *praeiti* "to pass"):

Praėjęs *pro gerinimo įrenginius, vanduo pasidaro švarus.*
"**After passing** through the water treatment plant, the water becomes clean."

Another example is a participle *patyręs* "experienced". It can be considered an antonym of the adjective *jaunas* "young" in the following case:

Kovą iškils finansinių sunkumų, bet dėl jų su niekuo nesitarkite – net su ***patyrusiais*** *specialistais.*
"There will be financial difficulties in March, but don't discuss them with anyone – even **experienced** professionals. "

However, there are cases where the participle *patyręs*, used in a different context, cannot be considered an antonym of the adjective *jaunas* "young":

Dar niekada, net ir vienišavimo laikais, nebuvau ***patyrusi*** *tokios vienatvės.*
"I have never **experienced** such loneliness, not even when I was single."

In conclusion, the search for adjectives and pronouns that are semantically similar to participles consisted of two phases:

1. Adjectives and pronouns were selected from the lists of distributionally similar words found using word embeddings (on fasttext.vdu.lt).
2. The linguist checked whether these selected adjectives and pronouns are semantically similar to the participles (i.e., used synonymously, anonymously, belonging to the same semantic classes). This was checked by analysing the concordances of participles in the Pedagogic Corpus in order to see the context of their usage.

3.2. Results

The analysis of the participles revealed that 48 of the 289 participles (16.61%) are semantically similar to adjectives or pronouns. Analysed participles with and without semantically similar adjectives or pronouns can be found in the appendices.

We observed that some participles are semantically similar to adjectives only when used with other words, e.g., *atrodantis* "looking" is used synonymously with the adjectives *estetiškas* "aesthetic", *simpatiškas* "handsome", *išvaizdus* "handsome", etc. only when used with certain adverbs of manner:

Esu kilnios širdies, liekna, ***gerai atrodanti****, nuoširdi, paprasta, draugiška*.
"I am noble-hearted, slender, **good-looking**, sincere, simple, friendly."

The expression *gerai atrodanti* can be replaced by adjectives *simpatiška* "attractive" and *išvaizdi* "good-looking".

Another example is the participle *dėtas* "put", used synonymously with the adjective *nekaltas* "innocent"/"accidental" when combined with the pronoun *niekuo* "by nobody":

Ponas Filypas iš peties vožtelėjo kumščiu per ***niekuo dėtą*** *kavos staliuką.*
"Mr Philip tapped his fist strongly on the **accidental** coffee table."

The expression *niekuo dėtą* can be replaced by the adjective *nekaltą* "innocent"/ "accidental".

The participle *tikęs* "suitable"/"good" can also be used synonymously with the participles *beviltiškas* "hopeless", *prasčiausias* "worst", *nevertas* "unworthy", etc., when used with the pronoun *niekam* "for nobody", for example:

<...> atrodo, jog dangus griūva, jog nieko nespėsite ir apskritai esate ***niekam tikę*** *šeimininkai.*
"<...> it seems that the sky is falling, that you won't be able to do anything in time, and that you are generally **useless** householders."

The expression *niekam tikę* can be replaced by adjectives *beviltiški* "hopeless", *prasčiausi* "the worst", and *neverti* "unworthy".

Although during this research 48 Lithuanian participles were identified as semantically similar to adjectives, only 23 of them are mentioned as synonyms or antonyms of Lithuanian adjectives in the Dictionary of Synonyms[8] and Dictionary of Antonyms[9].

The participles in this study, identified both by using word embeddings as semantically similar to adjectives and described in the Dictionary of Synonyms and Dictionary of Antonyms as having synonyms and antonyms of adjectives, are as follows:

- *atidarytas* "opened"/"open",
- *atliekamas* "leftover"/"spare",
- *gimtas* "native",
- *išgėręs* "having drunk",
- *keptas* "baked"/"fried",
- *matomas* "visible",
- *miręs* "dead",
- *nematomas* "invisible",
- *nematytas* "unseen",
- *nepamirštamas* "unforgettable",
- *nesuprantamas* "incomprehensible",
- *pastebimas* "noticeable",
- *praeitas* "last"/"previous",
- prieinamas "accessible"/"available"/"affordable",
- *priklausomas* "dependent",
- *privalomas* "mandatory",
- *suprantamas* "understandable",
- *teigiamas* "positive",
- *tikęs* "suitable"/"good",

[8] https://ekalba.lt/sinonimu-zodynas/
[9] https://ekalba.lt/antonimu-zodynas/

- *tinkamas* “suitable”/“appropriate”,
- *valgomas* “edible”,
- *virtas* “cooked”/“boiled”,
- *vykęs* “good”/“successful”.

Participles identified using word embeddings as semantically similar to adjectives, but not described in the Dictionary of Synonyms and Dictionary of Antonyms as having synonyms and antonyms of adjectives are as follows:

- *ateinantis* (*ateinančią savaitę* “next week”),
- *atrodantis* (*gerai atrodantis* “handsome”),
- *derantis* “matching”/“fitting”/“good”,
- *dėtas* (*niekuo dėtas* “innocent”/“accidental”),
- *dirbantis* “working”,
- *lankomas* (*lankomiausios vietos* “the most visited places”),
- *likęs* “remaining”,
- *mėgstamas* “likeable”,
- *mylimas* “dear”/“beloved”,
- *mylintis* “loving”,
- *naudojamas* “used”/“utilised”/“useful”,
- *naudotas* (*naudotas automobilis* “used car”),
- *nurodytas* “indicated”,
- *papildomas* “additional”/“extra”,
- *pasirinktas* “chosen”,
- *pastebėtas* “noticed”,
- *patyręs* “experienced”,
- *pažįstamas* “familiar”/“acquainted”/“known”,
- *praėjęs* “past”,
- *skirtas* (*kepimui skirtas indas* “a dish intended for baking”),
- *tikintis* “religious”,
- *tinkantis* “fitting”,
- *vadinamas* “called”/“referred to as”,
- *veikiantis* “functioning”/“working”,
- *verdantis* “boiling”.

On the one hand, this may be because these dictionaries include only the most characteristic synonyms and antonyms, whereas we identified not only synonyms and antonyms but also words that belong to the same semantic classes. On the other hand, these dictionaries omit some fairly typical synonyms and antonyms (e.g., *naudotas* “used” – *naujas* “new”; *papildomas* “additional” – *bazinis* “basic”, *būtinas* “necessary”).

It is also worth noting that the participle *priėjęs* is listed as a synonym of the adjective *brandus* (“ripe”) in the Dictionary of Synonyms, but it was not identified as synonymous with this adjective when using word embeddings. For example, in the phrase:

Priėję grūdai, vaisiai, uogos.
“Ripe grains, fruits, berries.”

However, this meaning of *priėjęs* is outdated and rarely used today, which may explain why it was not identified through word embeddings. The most common meaning of *priėjęs* is "having approached."

Thus, semi-automated, word-embedding-based methods can, in theory, identify more synonyms and antonyms than a linguist might detect through introspection or by consulting certain Lithuanian language resources.

4. Conclusions

The word embeddings approach can help identify adjectival participles by identifying adjectives and pronouns that are semantically similar to participles. Of the 289 analysed participles, 48 (16.61%) were found to be semantically similar to adjectives and pronouns.

This method is useful because it can help identify which participles are semantically similar to adjectives and pronouns, even though sometimes it can be difficult to identify these participles by introspection or by analysing other Lithuanian language resources. For instance, in the Dictionary of Synonyms and Dictionary of Antonyms, only 23 out of 48 participles are described as synonymous or anonymous to adjectives. This method is quite simple and provides more objective data, which would be time-consuming to obtain directly from a corpus. However, even though this method identified adjectival participles, it did not determine which of them had changed lexical meaning from the verb and lost the grammatical categories of participles that are not characteristic of adjectives (voice and tense). For this purpose, additional criteria for adjectival participles need to be applied.

It is important to note that analysis based on word embeddings requires a linguist's review of the results, which introduces an element of subjectivity at this stage of the research. In addition, this paper describes a study using the *fastText* model – it is likely that a different model may result in less noise in the analysis. Also, the training data should be of as high a quality as possible, for example, letters should be converted to lower case, and punctuation marks should be separated from words (this was not done in the present study). The results of the analysis also depend largely on the corpus – it is important to choose a sufficiently large and representative corpus in order to make the analysis as accurate as possible.

References

Altinok, D. (2024). Gender Bias in Turkish Word Embeddings: A Comprehensive Study of Syntax, Semantics and Morphology Across Domains, *in* Faleńska, A., Basta, C., Costa-jussà, M., Goldfarb-Tarrant, S., Nozza, D. (eds), *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, Association for Computational Linguistics, Bangkok, Thailand, pp. 203–218, available at https://aclanthology.org/2024.gebnlp-1.13.pdf.

Ambrazas, V. (ed.) (2006). *Lithuanian grammar*, second revised edition, Institute of the Lithuanian Language, Baltų lankų leidyba, Vilnius.

Benbaji, I., Doron, O., Hénot-Mortier, A. (2022). Word-Embeddings Distinguish Denominal and Root-Derived Verbs in Semitic, *in* Moortgat, M., Wijnholds, G. (eds), *Proceedings End-to-*

End Compositional Models of Vector-Based Semantics, 33rd European Summer School in Logic, Language and Information (15-16 Aug. 2022, NUI Galway, Galway, Ireland), EPTCS 366, 2022, pp. 35–49, available at https://arxiv.org/abs/2208.05721.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics* **5**, 135–146, available at https://aclanthology.org/Q17-1010.pdf.

Bumbulienė, I., Mandravickaitė, J., Bielinskienė, A., Boizou, L., Kovalevskaitė, J., Rimkutė, E., Vilkaitė-Lozdienė, L., Man, K. L., Krilavičius, T. (2018). RNNs for Lithuanian Multiword Expressions Identification, *International Journal of Design, Analysis and Tools for Integrated Circuits and Systems (IJDATICS)*, **7**(1), 44–47.

Firth, J. P. (1962). A synopsis of linguistic theory, 1930-1955, *in* Firth, J. et al. (eds), *Studies in Linguistic Analysis*, Blackwell, Oxford, pp. 1–32, available at https://cs.brown.edu/courses/csci2952d/readings/lecture1-firth.pdf.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T. (2018). Learning Word Vectors for 157 Languages, *in* Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, available at https://aclanthology.org/L18-1550/.

Gromann, D., Declerck, T. (2019). Towards the Detection and Formal Representation of Semantic Shifts in Inflectional Morphology, *in* Eskevich, M., de Melo, G., Fäth, Ch., McCrae, J. P., Buitelaar, P., Chiarcos, Ch., Klimek, B., Dojchinovski, M. (eds), *2nd Conference on Language, Data and Knowledge* (LDK 2019), LDK 2019 (10-23 May 2019, Leipzig, Germany), *Open Access Series in Informatics* (OASIcs), Vol. 70, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagsthul Publishing, Germany, pp. 21:1-21:15, available at https://doi.org/10.4230/OASIcs.LDK.2019.21.

Gutiérrez, L., Keith, B. (2019). A systematic literature review on word embeddings, *in* Mejia, J., Muñoz, M., Rocha, Á., Peña, A., & Pérez-Cisneros, M. (eds), *Trends and Applications in Software Engineering*, CIMPS 2018 (Oct 17-18 2018, Guadalajara, Jalisco, México), *Advances in Intelligent Systems and Computing*, Vol. 865, Springer, Cham, pp. 132–141, available at https://doi.org/10.1007/978-3-030-01171-0_12.

Haley, C., Ponti, E. M., Goldwater, S. (2023). Language-Agnostic Measures Discriminate Inflection and Derivation, *in* Beinborn, L., Goswami, K., Muradoğlu, S., Sorokin, A., Kumar, R., Shcherbakov, A., Ponti, E. M., Cotterell, R., Vylomova, E. (eds), *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, EACL 2023 (May 2-6, 2023, Dubrovnik, Croatia), Association for Computational Linguistics, Dubrovnik, Croatia, pp. 150-152, available at https://aclanthology.org/2023.sigtyp-1.18/.

Harris, Z. S. (1954). Distributional Structure, *Word* **10**(2-3), 146-162, available at https://doi.org/10.1080/00437956.1954.11659520.

Hong, Y., Lepage, Y. (2018). Production of Large Analogical Clusters from Smaller Example Seed Clusters Using Word Embeddings, *in* Cox, M., Funk, P., Begum, S. (eds), *Case-Based Reasoning Research and Development International*, ICCBR 2018 (10-12 Jul. 2018, Stockholm, Sweden), *Lecture Notes in Computer Science()*, Vol. 11156, Springer, Cham, pp. 548-562, available at https://doi.org/10.1007/978-3-030-01081-2_36.

Jancaitė, L. (2023). Subūdvardėjusių lietuvių kalbos dalyvių atpažinimo kriterijai (Criteria for Identifying Adjectival Participles in Lithuanian), *Taikomoji kalbotyra* **20**, 183-207, available at https://doi.org/10.15388/Taikalbot.2023.20.14.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification, *in* Lapata, M., Blunsom, P., Koller, A. (eds), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, EACL 2017 (April 3-7, 2017, Valencia, Spain), Association for

Computational Linguistics, Valencia, Spain, pp. 427-431, available at https://aclanthology.org/E17-2068/.

Jurafsky, D., Martin, J. H. (2024). *Speech and Language Processing*, third Edition draft of January 12, 2025, available at https://web.stanford.edu/~jurafsky/slp3/ed3book_Jan25.pdf.

Kapočiūtė-Dzikienė, J., Damaševičius, R., Woźniak, M. (2019). Sentiment Analysis of Lithuanian Texts Using Traditional and Deep Learning Approaches, *Computers* **8**, 4, available at https://doi.org/10.3390/computers8010004.

Kastner, I. (2020). Predicting semantic priming in Hebrew morphology using word embeddings, in Poster presented at *AMLaP* (*Architectures and Mechanisms for Language Processing*, Sep. 3-5, 2020, online conference).

Kovalevskaitė, J., Boizou, L., Bielinskienė, A., Jancaitė, L., Rimkutė. E. (2022). The First Corpus-Driven Lexical Database of Lithuanian as L2, *in* Utka, A. Vaičenonienė, J., Kovalevskaitė, J., Kalinauskaitė, D. (eds), *Human Language Technologies – The Baltic Perspective*, Proceedings of the Ninth International Conference Baltic HLT 2020 (Sep. 22-23, 2020, Kaunas, Lithuania), IOS Press, Amsterdam, Berlin, Washington, DC, pp. 245-252, available at https://ebooks.iospress.nl/doi/10.3233/FAIA200630.

Kovalevskaitė, J., Rimkutė, E., Vaičenonienė, J. (2021). Arbitraliųjų lietuvių kalbos kolokacijų nustatymas (Identification of Lithuanian Arbitrary Collocations), *Bendrinė kalba* **94**, available at https://etalpykla.lituanistika.lt/fedora/objects/LT-LDB-0001:J.04~2021~1662992740098/datastreams/DS.002.0.01.ARTIC/content.

Kutuzov, A., Velldal, E., Øvrelid, L. (2016). Redefining part-of-speech classes with distributional semantic models, *in* Riezler, S., Goldberg, Y. (eds), *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (Aug. 11-12, 2016 Berlin, Germany), The Association for Computational Linguistics, Berlin, Germany, pp. 115-125, available at https://aclanthology.org/K16-1012.pdf.

Lassner, D., Brandl, S., Baillot, A., Nakajima, S. (2023). Domain-Specific Word Embeddings with Structure Prediction, *Transactions of the Association for Computational Linguistics* **11**, 320-335, available at https://doi.org/10.1162/tacl_a_00538.

Naseem, U., Razzak, I., Khan, S. K., Prasad, M. (2021). A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models, *ACM Transactions on Asian and Low-Resource Language Information Processing* **20**(5), 1-35, available at https://doi.org/10.1145/3434237.

Paulauskienė, A. (1994). *Lietuvių kalbos morfologija: paskaitos lituanistams (The morphology of the Lithuanian language: Lectures to specialists of Lithuanian)*, Mokslo ir enciklopedijų leidykla, Vilnius.

Petkevičius, M., Vitkutė-Adžgauskienė, D. (2021). Intrinsic Word Embedding Model Evaluation for Lithuanian Language Using Adapted Similarity and Relatedness Benchmark Datasets, *in* Veitaitė, I., Lopata, A., Krilavičius, T., Woźniak, M. (eds), *CEUR workshop proceedings: IVUS 2021: proceedings of the 26th international conference on information society and university studies* (April 23, 2021, Kaunas, Lithuania), vol. 2915, CEUR-WS, Kaunas, Lithuania, available at https://ceur-ws.org/Vol-2915/paper14.pdf.

Petrunina, U. (2021). *Adjectivization in Russian: Analyzing participles by means of lexical frequency and constraint grammar*, PhD thesis, The Arctic University of Norway, Norway, available at https://munin.uit.no/bitstream/handle/10037/20757/thesis.pdf?sequence=2&isAllowed=y.

Rokas, A., Rackevičienė, S., Utka, A. (2020). Automatic extraction of Lithuanian cybersecurity terms using deep learning approaches, *in* Utka, A., Vaičenonienė, J., Kovalevskaitė, J., Kalinauskaitė, D. (eds), *Human Language Technologies – The Baltic Perspective*, Proceedings of the Ninth International Conference Baltic HLT 2020 (Sep. 22-23, 2020, Kaunas, Lithuania), IOS Press, Amsterdam, Berlin, Washington, DC, pp. 39-46, available at https://ebooks.iospress.nl/volumearticle/55521.

Stankevičius, L., Lukoševičius, M. (2020). Testing pre-trained transformer models for Lithuanian news clustering, *in* Lopata, A., Sukackė, V., Krilavičius, T., Veitaitė, I., Woźniak, M. (eds), *CEUR workshop proceedings: IVUS 2020: proceedings of the information society and university studies* (April 23, 2020, Kaunas, Lithuania), vol. 2698, CEUR-WS, Kaunas, Lithuania, pp. pp. 46-53, available at https://arxiv.org/abs/2004.03461.

Valeckienė, A. (n.d.). Įvardžiuotinės formos (Pronominal forms), *Visuotinė lietuvių enciklopedija* (Universal Lithuanian Encyclopedia), Mokslo ir enciklopedijų leidybos centras, available at https://www.vle.lt/straipsnis/ivardziuotines-formos/.

Wang, Y., Huang, G., Li, J., Li, H., Zhou, Y., Jiang, H. (2021). Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis, *IEEE Access* **9**, 37075-37085, available at https://doi.org/10.1109/ACCESS.2021.3062654.

Received January 31, 2025, revised March 9, 2025, accepted March 10, 2025

Appendix A

Analysed participles without semantically similar adjectives or pronouns

Aplankęs (“having visited”), *atėjęs* (“having come”), *atidaręs* (“having opened”), *atidaromas* (“being opened”), *atiduotas* (“given away”), *atliekantis* (“performing”), *atlikęs* (“having performed”), *atliktas* (“performed”), *atsakęs* (“having answered”), *atsakomas* (“being answered”), *atsiradęs* (“having appeared”), *atvažiavęs* (“having arrived (by vehicle)”), *atvažiuojantis* (“arriving (by vehicle)”), *atvykęs* (“having arrived”), *atvykstantis* (“arriving”), *augantis* (“growing”), *augęs* (“having grown”), *auginamas* (“being grown”), *auginantis* (“growing (something)”), *baigęs* (“having finished”), *baigiamas* (“being finished”), *baigtas* (“finished”), *bandomas* (“being tried/tested”), *dainuojantis* (“singing”), *dalyvaujantis* (“participating”), *dalyvavęs* (“having participated”), *darytas* (“made”), *daromas* (“being made”), *dedamas* (“being put”), *dirbamas* (“being worked”), *dirbęs* (“having worked”), *duotas* (“given”), *einantis* (“going”), *ėjęs* (“having gone”), *gaminamas* (“being made/produced”), *gaminantis* (“making/producing”), *gaunamas* (“being received”), *gautas* (“received”), gavęs (“having received”), *geriamas* (“being drunk”), *gimęs* (“born”), *girdėjęs* (“having heard”), *gyvenamas* (“inhabited/lived in”), *gyvenantis* (“living”), *gyvenęs* (“having lived”), *gyventas* (“lived”), *grįžęs* (“having returned”), *ieškantis* (“searching”), *ieškomas* (“being searched for”), *imamas* (“being taken”), *imtas* (“taken”), *įrengtas* (“installed”), *įsigijęs* (“having acquired”), *įsigytas* (“acquired”), *įsikūręs* (“having settled”), *išeinantis* (“leaving”), *išėjęs* (“having left”), *išgirdęs* (“having heard”), *išlikęs* (“having remained”), *išvykęs* (“having departed”), *jaučiamas* (“being felt”), *kalbamas* (“spoken”), *kalbantis* (“speaking”), *keičiamas* (“being changed”), *keliamas* (“being raised”), *keliantis* (“raising”), *keliaujantis* (“traveling”), *kepamas* (“being baked”), *ketinamas* (“intended”), *ketinantis* (“intending”), *kylantis* (“rising”), *kilęs* (“having risen/originated”), *kuriamas* (“being created”), *kuriantis* (“creating”), *kurtas* (“created”), *kviečiamas* (“being invited”), *laikantis* (“holding”), *laikytas* (“held”), *laikomas* (“being held/considered”), *lankantis* (“visiting”), *laukiamas* (“awaited”), *laukiantis* (“waiting”), *lauktas* (“waited for”), *leidžiamas* (“being allowed”), *leidžiantis* (“allowing/descending”), *manytas* (“thought”), *manomas* (“believed/assumed”), *matęs* (“having seen”), *matytas* (“seen”), *mėgstantis* (“liking”), *miegantis* (“sleeping”), *miegamas* (“being slept in”), *mokamas* (“paid”), *mokantis* (“knowing/able to”), *nematęs* (“not having seen”), *nešantis* (“carrying”), *nuėjęs* (“having gone (on foot)”), *nurodantis* (“indicating”), *nurodomas* (“being indicated”), *nusipirkęs* (“having bought (for oneself)”), *nusprendęs* (“having decided”), *nuspręstas* (“decided”), *nustatytas* (“set”, “established”), *nustatomas* (“being determined”), *padaręs* (“having done”), *padarytas* (“done”), *padaromas* (“being done”), *padedamas* (“being helped/placed”), *padedantis* (“helping”), *padėjęs* (“having placed/helped”), *padėtas* (“placed”), *paėmęs* (“having taken”), *pakeistas* (“changed”), *pakeitęs* (“having changed”), *pakėlęs* (“having raised”), *pakeliamas* (“being raised”), *pakeltas* (“raised”), *pakilęs* (“having risen”), *pakviestas* (“invited”), *pakvietęs* (“having invited”), *palikęs* (“having left”), *paliktas* (“left”), *pamatęs* (“having seen”), *pamiršęs* (“having forgotten”), *parašęs* (“having written”), *parašytas* (“written”), *parduodamas* (“being sold”), *parduotas* (“sold”), *parodytas* (“shown”), *paruoštas* (“prepared”), *pasakęs* (“having said”), pasakytas (“said”), *pasakojamas* (“being told”), *pasiekęs* (“having achieved”), *pasiekiamas* (“reachable”), *pasiėmęs* (“having taken (for oneself)”), *pasirenkamas* (“being chosen”), *pasirinkęs* (“having chosen”), *pasirinktas* (“chosen”), *pasirodęs* (“having appeared”), *pasiūlytas* (“offered”), *paskambinęs* (“having called”), *pastatęs* (“having built”), *pastatytas* (“built”), *pastebėjęs* (“having noticed”), *pastebėtas* (“noticed”), *patariamas* (“being advised”), *pateikęs* (“having presented”), *pateikiamas* (“being presented”), *pateiktas* (“presented”), *patekęs* (“having reached”), *patiekiamas* (“being served”), *patiektas* (“served”), *patirtas* (“experienced”), *perkamas* (“being bought”), *pirktas* (“bought”), *planuojamas* (“being planned”), *pradedamas* (“being started”), *pradedantis* (“starting”), *pradėjęs* (“having started”), *pradėtas* (“started”), *praleidžiantis* (“spending (time)”), *pranešamas* (“being announced”), *praneštas* (“announced”), *prasidėjęs* (“having begun”), *priėjęs* (“having

approached"), *priėmęs* ("having accepted"), *priimamas* ("being accepted"), *priimtas* ("accepted"), *priklausantis* ("belonging"), *priklausęs* ("having belonged"), *primenantis* ("reminding"), *prisiminęs* ("having remembered"), *radęs* ("having found"), *randamas* ("being found"), *rastas* ("found"), *rašęs* ("having written"), *rašomas* ("being written"), *reiškiantis* ("meaning/expressing"), *rengiamas* ("being prepared"), *renkamas* ("being collected/elected"), *rodantis* ("showing"), *rodomas* ("being shown"), *sakęs* ("having said"), *sakomas* ("being said"), *saugantis* ("protecting"), *saugomas* ("being protected"), *sėdintis* ("sitting"), *siekiamas* ("being aimed for"), *siekiantis* ("aiming for"), *siūlantis* ("offering"), *siūlomas* ("being offered"), *skaitęs* ("having read"), *skiriamas* ("being assigned"), *skiriantis* ("distinguishing"), *spaudžiamas* ("being pressed"), *statytas* ("built"), *statomas* ("being built"), *stebintis* ("observing"), *stovėjęs* ("having stood"), *stovintis* ("standing"), *sudarantis* ("forming/constituting"), *sudarytas* ("formed"), *sudaromas* ("being formed"), *sudedamas* ("being put together"), *sudėtas* ("put together"), *sukėlęs* ("having caused"), *sukeliantis* ("causing"), *sukeltas* ("caused"), *sukūręs* ("having created"), *sukurtas* ("created"), *sulaukęs* ("having received/awaited"), *supjaustytas* ("sliced"), *supjaustomas* ("being sliced"), *supratęs* ("having understood"), *suradęs* ("having found"), *susipažinęs* ("having become acquainted"), *susitikęs* ("having met"), *suteikiamas* ("being granted"), *suteikiantis* ("granting"), *suteiktas* ("granted"), *sutikęs* ("having agreed/met"), *sutiktas* ("agreed/met"), *sužinojęs* ("having learned"), *tapęs* ("having become"), *tekęs* ("having fallen to (one's lot)"), *tikimas* ("being believed"), *tikimasi* ("it is expected"), *vadintas* ("called"), *vaikščiojantis* ("walking around"), *vaikštantis* ("walking"), *vartojamas* ("being used"), *vartojantis* ("using"), *važiavęs* ("having traveled (by vehicle)"), *važiuojantis* ("traveling (by vehicle)"), *veikęs* ("having worked/functioned"), *veikiamas* ("being influenced"), *verdamas* ("being boiled"), *vertinamas* ("being evaluated"), *vertinantis* ("evaluating"), *vykstantis* ("happening").

Appendix B

Analysed participles (in alphabetical order) with semantically similar adjectives or pronouns

Participles	Semantically similar adjectives and pronouns
ateinantis (“upcoming”, “approaching”, “next”)	*dabartinis* (“current”), *artimiausias* (“closest,” “nearest”), *paskutinis* (“last”, “final”), *šis* (“this” - pronoun), *šitas* (“this one” - pronoun), *anasis* (“that one (farther)” - pronoun), *tas* (“that” - pronoun)
atidarytas (“opened”, “open”)	*atdaras* (“open”)
atliekamas (“leftover”, “spare”)	*laisvas* (“free”, “available”), *papildomas* (“additional”, “extra”)
atrodantis (“appearing”, “seeming”), *gerai atrodantis* (“handsome”)	*elegantiškas* (“elegant”), *simpatiškas* (“sympathetic,” “likeable”), *išvaizdus* (“handsome”, “presentable”), *žavingas* (“charming”), *nesimpatiškas* (“unlikeable”), *išraiškingas* (“expressive”), *neišvaizdus* (“plain”, “unattractive”), *estetiškas* (“aesthetic”), *stilingas* (“stylish”), *dailus* (“pretty,” “neat”)
derantis (“matching”, “fitting”, “good”)	*panašus* (“similar”), *identiškas* (“identical”)
dėtas (“placed”, “blamed”), *niekuo dėtas* (“innocent”, “accidental”)	*nekaltas* (“innocent”), *kaltas* (“guilty”)
dirbantis (“working”)	*bedarbis* (“unemployed”), *kvalifikuotas* (“qualified”), *darbingas* (“able to work”)
gimtas (“native”), *gimtas miestas* “hometown”	*svetimas* (“foreign”), *tolimas* (“distant”)
išgėręs (“having drunk”)	*blaivus* (“sober”), *neblaivus* (“intoxicated”), *girtas* (“drunk”), *girtutėlis* (“very drunk”)
keptas (“baked”, “fried”)	*šviežias* (“fresh”)
lankomas (“visited”), *lankomiausios vietos* (“the most visited places”)	*įdomiausias* (“most interesting”), *smagiausias* (“most fun”), *linksmiausias* (“funniest”), *mieliausias* (“most charming”), *populiariausias* (“most popular”), *įspūdingiausias* (“most impressive”), *garsiausias* (“most famous”), *vaizdingiausias* (“most picturesque”), *nykiausias* (“most boring/dull”)
likęs (“remaining”)	*didysis* (“major”), *mažytis* (“minor”, “tiny”), *žymus* (“significant”)
matomas (“visible”)	*tamsus* (“dark”), *akivaizdus* (“evident”, “obvious”), *raiškus* (“clear”, “distinct”), *vaizdus* (“vivid”, “graphic”)
mėgstamas (“liked”, “favorite”, “likeable”)	*geras* (“good”), *prastas* (“poor”), *kokybiškas* (“high-quality”), *žymiausias* (“most notable”), *populiariausias* (“most popular”), *įžymiausias* (“most famous”), *šūdiniausias*[10] (“worst”, “most awful”), *įdomiausias* (“most interesting”), *žaviausias* (“most charming”), *smagiausias* (“most fun”)

[10] The lexicon of informal language was analysed, e.g., *šūdinas* “shitty”, *fainas* “cool”, as such words were found when analysing word embeddings.

mylimas (“loved”, “dear”)	*mielas* (“dear”, “sweet”), *artimas* (“close”), *savasis* (“one's own”), *nesavas* (“alien”, “not one's own”), *meilus* (“affectionate”), *nuostabus* (“wonderful”), *gerasis* (“the good one”), *fainiausias* (“coolest”), *žavingiausias* (“most charming”), *šauniausias* (“nicest”)
mylintis (“loving”)	*rūpestingas* (“caring”), *egoistiškas* (“selfish”), *nedėkingas* (“ungrateful”), *motiniškas* (“motherly”), *ištikimas* (“loyal”), *doras* (“honest”), *nesavanaudis* (“selfless”), *dorovingas* (“virtuous”)
miręs (“dead”)	*gyvas* (“alive”), *negyvas* (“lifeless”), *nebegyvas* (“no longer alive”)
naudojamas (“used”, “utilized”, “useful”)	*reikalingas* (“necessary”)
naudotas (“used”, “second-hand”)	*senesnis* (“older”), *antrinis* (“secondary”), *antikvarinis* (“antique”), *naujesnis* (“newer”)
nematomas (“invisible”)	*tamsiausias* (“darkest”), *paslaptingas* (“mysterious”)
nematytas (“unseen”)	*neįtikėtinas* (“unbelievable”)
nepamirštamas (“unforgettable”)	*įprastas* (“usual”), *neįprastas* (“unusual”), *savotiškas* (“peculiar”), *nesėkmingas* (“unsuccessful”), *džiugus* (“joyful”), *neįtikėtinas* (“unbelievable”), *reikšmingas* (“significant”), *įdomiausias* (“most interesting”)
nesuprantamas (“incomprehensible”)	*keistas* (“strange”), *banalus* (“banal”), *elementarus* (“elementary”), *neesminis* (“non-essential”), *niekinis* (“insignificant”), *kvailas* (“stupid”), *neprotingas* (“unreasonable”), *painus* (“confusing”), *idiotiškas* (“idiotic”), *beprasmis* (“meaningless”)
nurodytas (“indicated”)	*konkretus* (“specific”), *tikslus* (“precise”), *netikslus* (“imprecise”)
papildomas (“additional”, “extra”)	*bazinis* (“basic”), *būtinas* (“necessary”), *specialus* (“special”)
pasirinktas (“chosen”)	*tikslingas* (“purposeful”)
pastebimas (“noticeable”)	*ryškesnis* (“more distinct”), *tamsiausias* (“darkest”), *geriausias* (“best”), *populiariausias* (“most popular”), *įdomiausias* (“most interesting”), *puikiausias* (“excellent”)
pastebėtas (“noticed”)	*įdomus* (“interesting”), *reikšmingas* (“significant”), *akivaizdus* (“evident”)
patyręs (“experienced”)	*jaunas* (“young”), *veiksnus* (“capable”), *stiprus* (“strong”), *neprofesionalus* (“unprofessional”), *pajėgus* (“able”)
pažįstamas (“familiar”, “acquainted”, “known”)	*artimas* (“close”), *draugiškas* (“friendly”), *šeimyniškas* (“family-like”), *nedraugiškas* (“unfriendly”), *neaiškus* (“unclear”)
praeitas (“passed”, “last”, “previous”)	*senesnis* (“older”), *paskutinis* (“last”), *anas* (“that one (past)”), *šis* (“this one” - pronoun), *kitas* (“another” - pronoun)
praėjęs (“past”)	*šiemetinis* (“this year's”), *ankstesnis* (“earlier”), *šiųmetis* (“of this year”), *pastarasis* (“recent”), *šiandieninis* (“today's”), *paskutinis* (“last”), *šis* (“this” - pronoun), *kažkuris* (“some one” - pronoun), *tas* (“that” – pronoun)
prieinamas (“accessible”, “available”, “affordable”)	*paprastas* (“simple”), *pigus* (“cheap”), *patogus* (“convenient”, “comfortable”)
priklausomas (“dependent”)	*savarankiškas* (“independent”), *laisvas* (“free”)

privalomas (“mandatory”)	*privalus* (“obligatory”), *priverstinis* (“forced”), *būtinas* (“necessary”)
skirtas (“intended”, “meant”), *kepimui skirtas indas* (“a dish intended for baking”)	*reikalingas* (“necessary”), *naudingas* (“useful”)
suprantamas (“understandable”)	*keistas* (“strange”), *keistokas* (“somewhat strange”), *mistinis* (“mystical”), *painus* (“confusing”), *akivaizdus* (“obvious”), *apgaulingas* (“misleading”)
teigiamas (“positive”)	*reikšmingas* (“significant”), *negatyvus* (“negative”), *geras* (“good”), *pozityvus* (“positive”), *blogas* (“bad”), *pesimistinis* (“pessimistic”), *viltingas* (“hopeful”), *optimistinis* (“optimistic”), *žalingas* (“harmful”), *negeras* (“not good”)
tikintis (“believing”, “faithful”, “religious”)	*bedieviškas* (“godless”), *religingas* (“religious”), *bedievis* (“atheist”), *nereligingas* (“non-religious”), *krikščioniškas* (“Christian”)
tikęs (“suitable”, “good”), *niekam tikęs* (“useless”, “worthless”)	*neįdomus* (“uninteresting”), *beviltiškas* (“hopeless”), *bukas* (“dull”), *bevertis* (“worthless”), *banalus* (“banal”), *menkavertis* (“low-value”), *nepraktiškas* (“impractical”), *prasčiausias* (“worst”), *profesionalus* (“professional”), *kvailas* (“foolish”)
tinkamas (“suitable”, “appropriate”)	*geras* (“good”), *idealus* (“ideal”), *pusėtinas* (“average”), *praktiškas* (“practical”), *maloniausias* (“most pleasant”), *puikus* (“excellent”), *reikalingas* (“necessary”)
tinkantis (“fitting”), *jums tinkanti šukuosena* (“the hairstyle that suits you”)	*geriausias* (“best”), *idealus* (“ideal”), *maloniausias* (“most pleasant”), *patogus* (“comfortable”), *reikalingas* (“necessary”)
veikiantis (“functioning”, “working”)	*aktyvus* (“active”), *funkcionalus* (“functional”), *efektyvus* (“effective”)
vadinamas (“called”, “referred to as”)	*tas* (“that”), *savotiškas* (“peculiar”), *kažkoks* (“some kind of” - pronoun)
valgomas (“edible”)	*nuodingas* (“poisonous”)
verdantis (“boiling”)	*karštas* (“hot”)
vykęs (“successful”, “good”)	*sėkmingas* (“successful”), *nesėkmingas* (“unsuccessful”), *įdomus* (“interesting”), *savotiškas* (“peculiar”), *geriausias* (“best”)
virtas (“boiled”, “cooked”)	*šviežias* (“fresh”), *žalias* (“raw”)

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 252–289
https://doi.org/10.22364/bjmc.2025.13.1.14

Real-Time Phone Fraud Detection and Prevention Based on Artificial Intelligence Tools

Roberts OĻEIŅIKS, Darja SOLODOVŅIKOVA

Faculty of Computing, University of Latvia, Riga, Latvia

olein.roberts@gmail.com, darja.solodovnikova@lu.lv

ORCID 0009-0009-6867-1357, ORCID 0000-0002-5585-2118

Abstract. Telephone fraud poses significant threats to telecommunications network users, causing both financial loss and emotional stress. The aim of this study was to develop and evaluate a real-time telephone fraud detection and prevention system, based on the principle of phone conversation content analysis. The study included an empirical study to select optimal AI tools for system implementation. A system prototype was developed, integrating the selected automatic speech recognition tool and large language model with a specific prompt, to analyze phone conversation content in real-time. The system's effectiveness was evaluated in a simulated environment reflecting real-time conditions, using an expanded dataset with various fraud scenarios and languages.

The results indicate high classification effectiveness of the system, achieving an accuracy of 90,4% and a 91,2% F1 score, indicating the system's efficacy in real-time telephone fraud detection. The prevention rate reached 69,8%, demonstrating the system's potential in real-time telephone fraud prevention

Keywords: real-time telephone fraud detection, artificial intelligence (AI), phone conversation content analysis, real-time telephone fraud prevention, large language model (LLM), automatic speech recognition (ASR)

1 Introduction

The evolution of telecommunications technology has significantly transformed daily life by enabling rapid and efficient communication both personally and professionally. Telecommunications have not only made the exchange of calls and messages quicker and more convenient but also opened new avenues for remote work and education, allowing people to communicate and collaborate irrespective of their physical location.

However, despite many benefits, telecommunications technology has also introduced new security challenges, including telephone fraud. This type of fraud, which aims to generate illegal income using telecommunications, presents significant threats

affecting all network users, impacting individual consumers and legal entities such as mobile network operators. Although protecting mobile network operators from fraudsters is crucial, this research primarily focuses on individual end-users, including private individuals and company employees, who may encounter phone fraudsters daily. Studies highlight that while there are strategies to combat fraud targeting mobile network operators (Sahin et al., 2017; Trapiņš, 2015), individual users are still vulnerable, as shown by telephone fraud statistics.

Surveys by the European Commission in 2020 (WEB, a) estimated that, over two years, European citizens lost EUR 24 billion EUR due to fraud, with 28% of these fraud cases conducted via telephone. In 2023, the U.S. Federal Trade Commission reported that telephone fraud cost consumers approximately 850 million USD (WEB, b). In Latvia, the financial industry association noted that phone scams are challenging to prevent and phone fraud cases resulted in losses of 5,5 million EUR for Latvian citizens in 2023 alone (WEB, c).

Official statistics on telephone fraud do not fully capture the problem's scope. Victims often do not report their experiences, out of shame or fear, suggesting that the true financial losses could be considerably higher. The psychological and emotional toll on victims of telephone fraud, including stress, fear, anger, and psychological trauma, indicates a deeper impact on both individual and societal levels beyond mere financial losses (WEB, a,d). Overall, the statistics not only highlight the problem's relevance but also the potential need for new and effective protection methods in this field.

The structure of this paper is organized as follows: Section 2 outlines the research aim, establishing the study's primary objectives and framework. Section 3 covers related work, reviewing existing literature and technologies relevant to phone fraud detection and prevention. Section 4 presents the system prototype design, detailing the architecture of the real-time phone fraud detection and prevention system, which utilizes AI tools such as Automatic Speech Recognition (ASR) and Large Language Models (LLMs). Section 5 presents study, which focuses on choosing the most suitable ASR tool to transcribe conversations, while Section 6 describes the selection process for an LLM to analyze conversations for fraud detection. In Section 7, an empirical evaluation of the integrated system is presented, assessing its accuracy, latency, and real-time capabilities. Section 8 provides a discussion on the findings, and Section 9 concludes the study, summarizing key insights and suggesting directions for future work.

2 Research aim

This study addresses the critical need for a real-time phone fraud detection and prevention system tailored for individual users within telecommunications networks.

The goal of this research is to design and evaluate a full-scale real-time telephone fraud detection and prevention system, which aims to protect individual telecommunications network users from phone fraud. The system operates by analyzing phone conversation content by use of four main components: real-time recording of phone conversations, transcription, conversation content analysis, and user notification. It leverages existing AI tools for automatic speech recognition and large language model (LLM) to handle conversation transcription and content analysis.

The objectives of this research are threefold:

1. **To design an integrated system architecture** that incorporates its components effectively, ensuring seamless functionality and user responsiveness.
2. **To select the most suitable AI tools for transcription and content analysis**, focusing on achieving high accuracy and minimal latency in fraud detection.
3. **To conduct testing and simulations** to evaluate the system's effectiveness in real-world scenarios.

Further, empirical research will be conducted to optimize the selection of AI tools, with a comparative analysis of industry-provided technologies to determine the most effective ASR and LLM solutions. Additionally, the study will utilize a multi-lingual dataset of simulated phone conversations in Latvian, English, and Russian to mirror the linguistic context of Latvia, accompanied by a control group of legitimate phone calls for a comprehensive evaluation of the system's capabilities.

3 Related work

Phone fraud remains a significant and evolving threat to individuals and organizations, necessitating continuous advancements in detection and prevention methods. Existing approaches to combating phone fraud can be broadly categorized into two main categories: social education and technological solutions. Social education initiatives focus on raising public awareness about the characteristics of phone scams, empowering individuals to recognize and avoid potential fraud attempts. While valuable, such methods can be inconsistent, as human judgment is often susceptible to factors like fatigue or stress. This review focuses on technological methods for phone fraud detection, which is essentially a binary classification problem - distinguishing between legitimate and fraudulent calls. Evaluating the effectiveness of phone fraud detection systems involves considering various performance metrics such as accuracy, precision, recall and F1 score.

3.1 Phone call filtering and blocking

Traditionally, methods like blacklisting and filtering have been used to detect phone fraud. Call filtering methods involve marking suspicious numbers, timely blocking, and adding them to blacklists. Filtering and blocking are performed by analyzing Call Detail Records (CDR), which include call metadata — call start time, call end time, caller number, receiver number, call duration, receiver location, call type (outgoing, incoming), etc. In addition to call metadata, number tagging and blocking can be based on previous fraud cases and user reports. Implementations can use traditional data analysis algorithms or machine learning algorithms.

In a 2018 study, the "TouchPal" system was developed, employing machine learning methods and CDR data analysis (Li et al., 2018). The system relies on a substantial user base to create a reputation-based blacklist for effective prevention of fraudulent calls. Users mark suspicious calls, contributing to a blacklist that blocks incoming calls from these numbers. Machine learning models—including random forests, neural networks,

support vector machines, and logistic regression—predict fraudulent calls based on 29 features without analyzing call content. Key criteria include call type, duration, time, location, contact information, and historical data. The study reported a 99,99% precision rate and a 90% recall rate (calculated F1 score of 94,74%). However, the study notes that the main drawback is that fraudsters can bypass the system's protection by spoofing the caller ID.

The 2020 study by Xing et al. (Xing et al., 2020) presented a deep learning approach for automatic detection of fraudulent calls by analyzing CDRs. Models used included deep neural networks, convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and stacked denoising autoencoders (SDAEs). The SDAE model achieved over 99% accuracy. However, precision was low (6–11%) due to an imbalanced dataset (8,2 million legitimate calls vs. 8200 fraudulent calls). Given the low precision, the model avoided missing fraudulent calls by often misclassifying legitimate calls as fraudulent, sacrificing precision to ensure a high accuracy.

The 2021 study (Gowri et al., 2021) proposed a machine learning solution using a recurrent neural network (RNN) to detect malicious (fraudulent) phone calls by analyzing a dataset containing CDRs obtained from the Kaggle platform. The proposed solution includes data preprocessing for quality improvement and RNN model training for predictions. The study reported an 87% accuracy in detecting malicious calls, indicating the effectiveness of RNNs in reducing such calls. However, detailed performance metrics like recall, precision, and F1 score were not provided, limiting a comprehensive evaluation of the model's ability to distinguish between legitimate and fraudulent calls.

3.2 Caller ID spoofing

Filtering and blacklisting approaches are vulnerable to caller ID spoofing, which allows fraudsters to bypass protection mechanisms. Caller ID spoofing enables fraudsters to conceal their identity by altering their phone number to any other number. This is possible because telecommunication networks are not fully protected against such interference (Song et al., 2014). Initially, telecommunication systems were designed assuming that caller ID information would be authentic, unique, and authorized. However, technological advancements have allowed fraudsters to exploit system vulnerabilities, using software and devices that manipulate caller ID data. This manipulation allows fraudsters to generate any outgoing phone number, including ones similar to the victim's own number, contacts from the victim's phonebook, emergency numbers, etc. (WEB, e), making it harder to identify fraudulent calls and increasing the likelihood that the victim will answer. Caller ID spoofing reduces the effectiveness of filtering and blacklisting systems because these systems rely on historical data and user reports, which become useless when caller IDs are spoofed.

The 2020 study (Pandit et al., 2020) describes the development of the "RobocallGuard" system—a virtual assistant (VA) designed to limit fraudulent and other unwanted calls using automatic call filtering. Motivated by the problem of caller ID spoofing, "RobocallGuard" offers protection against the negative effects of phone number manipulation. The system uses audio analysis and voice recognition, serving as a protective layer between the caller and the recipient by filtering fraudulent and other unwanted calls using a test similar to a "captcha" but in audio format: the caller must state

the recipient's name. If the recipient's name is not correctly provided, the call is not connected and is subsequently blocked by adding it to a blacklist. If a call is received from a blacklisted number, the virtual assistant immediately blocks it; calls from the whitelist are connected without VA intervention. For numbers not in either list, the VA evaluates the call. The study shows a 97,8% recall rate but does not provide precision results. A significant limitation of the system is its inability to protect against targeted attacks where the caller knows the recipient's name. The VA's protection can be bypassed by using common names (e.g., "Peter", "Anna") or crafting phrases that might mislead the system by exploiting vulnerabilities in the voice recognition module.

3.3 Call content analysis

Traditional methods based largely on blacklisting are no longer sufficiently effective because such protection often does not offer comprehensive defense; fraudsters have found ways to bypass these systems by spoofing caller IDs. Moreover, if traditional methods allow a fraudster to establish direct contact with the victim, they are ineffective against social engineering techniques used during the call to manipulate the victim into facilitating fraud. The increasing use and availability of caller ID spoofing software have created new challenges in preventing phone fraud. Therefore, developing and implementing new methods based on analyzing the content of phone calls has become increasingly important. Such methods allow detecting phone fraud by analyzing the caller's speech content, sentiment, or other elements that may indicate fraudulent intent, such as social engineering techniques like imposing decision-making pressure and urgency. Thus, analyzing call content offers a significant step toward more effective detection of phone fraud, capable of addressing the new challenges posed by caller ID spoofing, social engineering techniques, and verbal manipulation during calls.

The early beginnings of call content analysis can be seen in the 2016 study (Sawa et al., 2016), where a method was developed using Natural Language Processing (NLP) to identify social engineering threats in text format, such as emails or social media chats. The method analyzes dialogue by focusing on questions and commands, comparing them to a predefined "blacklist" of topics. This list consists of forbidden actions toward specific objects and is tailored to specific situations. For example, an attacker wants the victim to manipulate a network device by saying "reset the router." This statement is identified as a potential threat because "reset" describes an action and "router" describes an object, matching an entry in the blacklist of forbidden topics. Due to the match, further communication is blocked, and a warning is sent to the victim. The study demonstrated a low false positive rate, 100% precision, and a 60% recall rate (calculated F1 score of 75%). A potential problem with this method is its dependence on the blacklist, which requires regular and manual updates—a time-consuming process that may not keep pace with evolving fraud tactics.

In contrast, the 2018 study (Zhao et al., 2018) fully presented a new approach to phone fraud detection based on analyzing call content, moving away from traditional methods. The study collected 12368 examples of fraudulent call descriptions from Chinese internet platforms like Sina Weibo and Baidu to create a dataset. The authors used NLP methods to extract features from texts, creating detection rules and then training

a model to perform text analysis. The model's effectiveness is highlighted by high prediction and performance metrics in the selected dataset—accuracy 98,53%, precision 97,97%, recall 98,25%, and F1 score 98,11%. An Android application was developed to implement the detection rules, offering real-time fraud detection without requiring user data upload to a server, thus ensuring privacy. To test the application's performance, experiments were conducted where participants were asked to read fraud dialogues. Out of 15 dialogues based on online resources and victim descriptions, 10 were in Mandarin, and 5 in dialects typical for China. The application detected 90% of fraud attempts in Mandarin but only 40% in dialects, indicating limitations in speech recognition technology. Despite the model's effectiveness, the study acknowledges limitations such as insufficient data volume and low accuracy of local speech recognition technology. It is important to note that due to Google's policy changes on April 6, 2022, two-way call recording was restricted on Android (WEB, f), making it difficult to assess whether the call recording method implemented in the study is still feasible and whether the overall system is operational.

The 2021 study (Derakhshan et al., 2021) developed a method to recognize social engineering techniques during phone calls using the innovative concept of "scam signatures", similar to malware signatures. Scam signatures are defined as sets of speech acts that form fraud indicators, serving as basic tools for a fraudster to achieve their goals. The study developed the Anti-Social Engineering Tool, which uses word and sentence embeddings from NLP to determine if scam signature elements are present in a conversation. The method demonstrates 100% precision, and a 71,4% recall rate (calculated F1 score of 83%).

The 2023 study (Hong et al., 2023) explored the use of LSTM model architecture in identifying fraudulent calls using call content. A balanced dataset of 100 call scenario recordings (50 fraudulent and 50 legitimate) was used, mainly obtained from YouTube. Call recordings were converted to text for further analysis. The model demonstrated moderate effectiveness, with an accuracy of 85,6%, precision of 60%, recall of 32%, and an F1 score of 41%. Despite promising accuracy, the overall system effectiveness is lower compared to other methods when evaluated using the F1 score. The study acknowledges limitations related to the dataset size and recommends improvements, including multilingual support, to enhance detection effectiveness.

3.4 Conclusion

Recognizing the limitations of current approaches, which primarily focus on fraud detection, this research contributes not only by developing a novel real-time system that integrates existing AI-based tools and methods for phone fraud detection but also by prioritizing the evaluation of its prevention capabilities, a crucial aspect overlooked in the literature.

4 System prototype design

This section presents the design and architecture of the proposed real-time phone fraud detection and prevention system prototype. The system leverages a content analysis

approach, utilizing AI tools such as Automatic Speech Recognition (ASR) and Large Language Models (LLMs) to identify and mitigate fraudulent calls in real-time.

4.1 System architecture

The system prototype is built upon four main components that work together to provide real-time fraud detection and prevention:

1. **Real-time phone call recording:** capture the audio stream of the phone conversation.
2. **Real-time call transcription:** convert the recorded audio into text using ASR.
3. **Real-time conversation content analysis:** analyze the transcribed text using LLMs to identify fraudulent patterns and tactics.
4. **Fraud notification:** alert the call recipient about detected fraud attempts.

The system operates by continuously monitoring the conversation content and triggering an alert if fraudulent activity is detected. Upon detection, the system can automatically terminate the call to prevent further interaction with the fraudster and send an SMS notification to the call recipient.

4.2 System design and modularity

The system prototype is built upon a modular architecture that integrates external services like Twilio, Google, OpenAI and nGrok, allowing for flexibility and adaptability in incorporating alternative or improved AI tools in the future. The system also incorporates a custom-developed middleware to manage data flow, audio processing, redaction, and user notifications, seamlessly connecting the various components and external services.

A key design consideration was ensuring compliance with data privacy regulations, particularly GDPR. The system utilizes a call forwarding mechanism to seamlessly connect incoming calls to the recipient's personal phone number while simultaneously recording the conversation through Twilio. To address GDPR requirements, the system can be configured to inform callers about the recording and obtain their consent before proceeding, aligning with legal requirements for lawful call recording. Additionally, personal identifiable information (PII) is redacted from transcripts before they are sent to the LLM for analysis, further protecting user privacy.

The real-time phone fraud detection and prevention system prototype outlines an innovative approach to addressing the evolving phone fraud threat by leveraging existing tools to provide AI-powered conversation content analysis. The detailed architectural design of the phone fraud detection and prevention system prototype is shown below in Figure 1. The phone call monitoring process operates continuously until either phone fraud is detected or the phone call ends. In a real-world deployment, the middleware is designed to be hosted on a cloud server, serving as the central hub between the telecommunications network and external services (e.g., Twilio, OpenAI, and Google). In this configuration, incoming calls are first forwarded via Twilio to the middleware, which

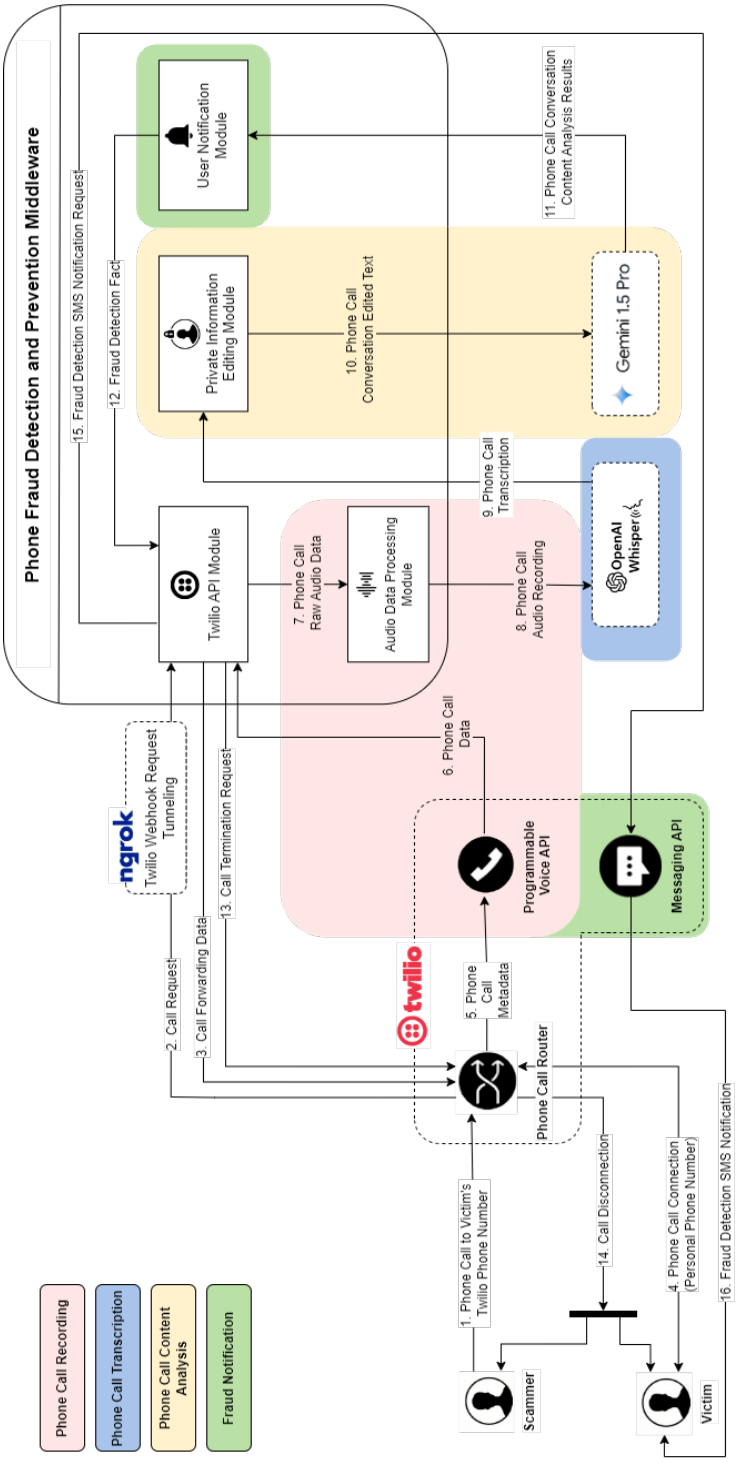


Fig. 1. Phone Fraud Detection and Prevention System Prototype Architecture Design

then manages data flow, audio processing, redaction, and user notifications. Communication traverses multiple network hops—from the mobile network to Twilio, then from Twilio to the middleware, and finally from the middleware to the external APIs.

This prototype serves as the foundation for subsequent sections, which detail the selection and evaluation of specific AI tools and the overall system performance.

4.3 Phone call recording

Workflow The call recording component is initiated when a user receives a phone call. The process involves the following steps (refer to Figure 1 for visual representation):

1. **Call initiation:** The caller dials the recipient's Twilio phone number.
2. **Webhook request:** Upon receiving the call, Twilio's call routing mechanism automatically generates an HTTP request to a pre-configured webhook address. This webhook address, associated with the recipient's Twilio number, acts as an interface between the call and the system's middleware.
3. **Call forwarding:** The Twilio API module within the middleware stores the recipient's personal phone number. Upon receiving the webhook request, it instructs Twilio to forward the call to the recipient's personal number.
4. **Call connection:** Twilio's call routing mechanism forwards the call to the recipient's personal phone number, establishing the connection between the caller and recipient.
5. **Metadata forwarding:** Once the call is connected, the call routing mechanism sends the call metadata to Twilio's Programmable Voice API to initiate call recording.
6. **Call data transmission:** Twilio's Programmable Voice API continuously sends call data, including metadata and raw audio, to the middleware's Twilio API module.
7. **Audio data preparation:** The middleware processes the raw audio data, separating and sequentially ordering the audio streams for the caller and recipient. Once approximately 15 seconds of audio data is accumulated, it is forwarded to the audio data processing module.
8. **Audio recording preparation:** The audio data processing module converts the raw audio data from Twilio's format (8-bit PCM mono) to MP3 format, which is compatible with the transcription tool. The module also merges the caller and recipient audio streams into a single MP3 file, which is then sent to the call transcription module.

4.4 Phone call transcription

The system utilizes OpenAI's "Whisper" speech recognition tool to transcribe the recorded phone calls into text. Whisper was selected based on empirical research comparing the performance of various speech recognition tools for the target languages (Latvian, English, and Russian). This research is discussed in subsequent chapter "5. Transcription tool selection".

Workflow

9. **Transcription:** The phone call recording, in MP3 format, is sent to OpenAI's Whisper API. Whisper processes the audio and generates a text transcript of the conversation, which is returned as the API response.

4.5 Conversation content analysis

The Google Gemini 1.5-Pro Large Language Model (LLM) is employed to analyze the transcribed conversation content and detect potential phone fraud. This LLM was chosen based on empirical research (discussed in chapter "6. Content analysis tool selection") evaluating the effectiveness of various LLM tools and parameter configurations across the target languages.

Workflow

10. **Text redaction:** The transcribed text is first processed by the private information redaction module. This module replaces personal identifiable information (PII), such as names and phone numbers, with placeholders to protect user privacy and comply with GDPR regulations.
11. **Content analysis:** The redacted transcript is sent to the Google Gemini 1.5-Pro LLM via its API for analysis. The LLM assesses the content and returns a judgment indicating whether the conversation excerpt exhibits characteristics of a phone fraud attempt. The results of the content analysis are forwarded to the user notification module.

4.6 Fraud notification

The system utilizes SMS notifications via Twilio's Messaging API to alert the call recipient about detected fraud attempts.

Workflow

12. **Fraud detection:** The user notification module receives the content analysis results from the Google Gemini LLM. If fraud is detected, the module immediately informs the Twilio API module.
13. **Call termination request:** Upon receiving the fraud detection alert, the Twilio API module sends a call termination request to Twilio to prevent further interaction between the recipient and the potential fraudster.
14. **Call termination:** Twilio's call routing mechanism terminates the call based on the metadata provided in the termination request.
15. **Notification request:** Simultaneously, the Twilio API module sends an SMS notification request to Twilio's Messaging API.
16. **SMS notification:** Twilio's Messaging API sends an SMS message to the call recipient's personal phone number, informing them of the detected fraud attempt.

Further research and experimental simulations in subsequent chapters will provide a deeper understanding of the system's operation and its ability to effectively detect and prevent phone fraud in real-time.

5 Transcription tool selection

In this section, an empirical study is conducted to select the most optimal transcription tool for use in a phone fraud detection and prevention system. The effectiveness of a real-time phone fraud detection and prevention system hinges on the ability to accurately and swiftly transcribe spoken conversations into text. This transcription process is critical, as it directly impacts the subsequent analysis and detection of fraudulent activities and indicators within the conversation. Inaccurate transcriptions can distort the content or context of conversations, impeding the system's ability to correctly identify fraudulent patterns. Similarly, delays in transcription can postpone the detection and prevention measures, increasing the likelihood of successful fraud attempts.

Despite the abundance of Automatic Speech Recognition (ASR) tools available today, there is a notable scarcity of recent benchmarking studies that evaluate their performance, especially in the context of real-time systems and for languages less commonly supported, such as Latvian. The rapid pace of technological advancement in ASR tools means that existing studies can quickly become outdated, underscoring the necessity for up-to-date evaluations.

Previous research has attempted to benchmark ASR systems. For instance, a study conducted in 2020 (Filippidou and Moussiades, 2020) evaluated several ASR systems, including Google, IBM Watson, and Wit.ai, using metrics such as Word Error Rate (WER), Hypothesis Error Rate (Hper), and Reference Position-Independent Word Error Rate (Rper). WER is a widely used metric for assessing the accuracy of speech recognition systems, which is calculated by comparing the original text with the transcribed text to determine the proportion of errors. The 2020 study found that Google's ASR system outperformed others, ranking first in terms of WER.

In contrast, a more recent study in 2023 (Ferraro et al., 2023) compared open-source ASR tools with commercial ones, using seven widely adopted datasets such as LibriSpeech and Common Voice. The open-source tools included Conformer, HuBERT, SpeechBrain, WhisperX, and SpeechStew, while the commercial tools were Amazon Transcribe, Microsoft Azure, Google, and IBM Watson. The results indicated that commercial ASR tools generally offered better performance in terms of accuracy and speed, although performance varied depending on the dataset. Notably, Amazon Transcribe and Microsoft Azure outperformed Google, suggesting a shift in the competitive landscape of ASR tools since the 2020 study.

Moreover, several unofficial articles and self-sponsored reports claim superior performance of other commercial ASR tools, such as DeepGram (WEB, g), AssemblyAI (WEB, h), Speechmatics (WEB, i), and RevAI (WEB, j). These claims often highlight impressive metrics, but the lack of independent verification necessitates an empirical evaluation to objectively assess their performance.

Given the evolving nature of ASR technology and the importance of accurate and rapid transcription in a phone fraud detection system, this part of the research aims to select the most suitable ASR tool in the context of phone fraud detection and prevention by conducting an empirical evaluation of several leading options. The tools initially considered for evaluation were Amazon Transcribe, Microsoft Azure, OpenAI Whisper, DeepGram, AssemblyAI, Speechmatics, and RevAI. These tools were selected based

on their prominence in previous studies, claimed performance metrics, and their availability for commercial use.

5.1 Methodology

The selection process was structured as a multi-stage empirical evaluation designed to simulate real-world conditions and focus on parameters critical to the system's performance. The methodology comprised the following stages:

1. Preselection and audio format impact analysis.
2. Benchmarking based on WER.
3. Benchmarking based on latency.

Stage 1: preselection and audio format impact analysis The initial stage comprised a qualitative assessment of language support and automatic language detection, followed by an analysis of how different audio formats impacted performance:

- Preselection: ASR tools were first evaluated based on their support for the three target languages (Latvian, English, and Russian) and their ability to automatically detect and transcribe conversations in the appropriate language without manual input. Tools that did not meet these criteria were excluded from further evaluation.
- Audio Format Impact: For the remaining tools, the effect of audio format (WAV vs. MP3) on transcription accuracy and latency was assessed. Four conversation scenarios (two fraudulent and two legitimate) were selected in each language for testing. WER was calculated for both formats to measure transcription accuracy, and latency was measured by recording the time it took to transcribe 15-second audio segments via API to simulate real-time processing.

Stage 2: benchmarking based on WER In this stage, the tools were evaluated based on their transcription accuracy using sample dataset of 10 scenarios. WER was calculated for each tool in all three languages. Tools that exhibited consistently high WER values across multiple languages were considered less suitable for real-time fraud detection, where high transcription accuracy is critical.

Stage 3: benchmarking based on latency The final stage assessed the latency of the ASR tools that performed well in terms of WER. Latency was measured using same sample dataset of scenarios by sending the audio files in 15-second segments to the ASR tool's API and recording the time taken to return the transcriptions. In a real-time fraud detection system, minimizing latency is essential for timely detection and prevention of fraudulent activities.

5.2 Dataset preparation

To evaluate the ASR tools under conditions reflective of real-world usage, a sample dataset was constructed. The dataset included 30 phone conversation scenarios, with 10 unique scenarios duplicated in each of the target languages—Latvian, English, and

Russian. Each language set comprised 5 fraudulent and 5 legitimate scenarios. More detailed information regarding dataset can be found in integrated system's evaluation chapter "7.1 Methodology".

The audio recordings for these conversation scenarios were generated using high-quality computer-generated voices from the "PlayHT" platform (WEB, k). Utilizing computer-generated voices allowed for precise control over speech parameters, particularly Words Per Minute (WPM), ensuring consistency across recordings. The WPM settings were chosen to reflect natural speaking speeds: 200 WPM for English, 170 WPM for Latvian, and 160 WPM for Russian (Rodero, 2012; Kappen et al., 2024; Bogdanovs, 2018), with a maximum variation of ±5 WPM.

To simulate the audio quality typically encountered in telecommunication networks, the original high-quality recordings were degraded to match standard telecommunication audio formats—8-bit PCM mono uLaw with an 8 kHz sampling rate. This step ensured that the ASR tools would be evaluated under realistic audio quality conditions that mirror actual phone conversations.

The degraded audio files were then converted into both WAV (lossless) and MP3 (lossy) formats.

5.3 Limitations and assumptions

The study acknowledged several limitations. The use of computer-generated voices may not fully capture the nuances of human speech, such as emotional variation or dialect, potentially affecting the accuracy of the transcription. The absence of background noise and telecommunication disruptions in the audio recordings might lead to an overestimation of transcription accuracy in real-world scenarios. Furthermore, latency measurements were conducted in a controlled environment, and real-world network conditions may vary, potentially affecting the system's performance.

The study also relied on several assumptions. It was assumed that all audio recordings were of uniform quality and that network conditions remained stable during testing, which may not always hold true in real-world applications.

5.4 Results

The ASR tool evaluation involved a total of 1439 latency measurements and 96 WER measurements in the first stage (audio format impact analysis), 150 WER measurements in the second stage, and 1615 latency measurements in the third stage.

Stage 1: Preselection and audio format impact analysis All ASR tools selected for the evaluation—Amazon Transcribe, Microsoft Azure, OpenAI Whisper, Speechmatics, RevAI, DeepGram, and AssemblyAI—demonstrated the ability to automatically identify the spoken language. Furthermore, Amazon Transcribe, Microsoft Azure, OpenAI Whisper, Speechmatics and RevAI provided full support for the target languages (Latvian, English, and Russian). However, DeepGram and AssemblyAI lacked support for Latvian and were excluded from further selection.

Based on these qualitative parameters, Amazon Transcribe, Microsoft Azure, OpenAI Whisper, RevAI, and Speechmatics were deemed suitable candidates for further quantitative evaluation.

Converting audio recordings from WAV to MP3 resulted in a substantial reduction in file size, with an average decrease of approximately 87,5%, given that the average conversation length was 4 minutes and 3 seconds. This reduction significantly impacts data transmission times, which is critical for a real-time system.

The transcription accuracy, measured by WER, showed a slight increase when using the MP3 format. The average WER, calculated across all selected ASR tools, increased by only 0,39% (from 5,62% to 6,01%), which is negligible in practical terms. On the other hand, processing latency decreased notably when using MP3. The average latency, calculated across all selected ASR tools, was reduced by 0,89 seconds (from 9,54 seconds with WAV to 8,65 seconds with MP3), representing a meaningful improvement for real-time processing.

Given the significant reduction in latency with only a minimal impact on transcription accuracy, the MP3 format was selected for subsequent evaluation stages. Microsoft Azure, which only supported WAV format at the time of testing, was not included in format evaluation.

Stage 2: Benchmarking based on word error rate Figure 2 presents a box plot illustrating the distribution of WER for each ASR tool in English. OpenAI's tool shows an average word error rate of 1,87%, with 50% of the data points falling between 0,85% and 2,65%. In contrast, Microsoft's tool exhibits a higher average WER of 5,54%, with an outlier at 11,60%. Based on English language recognition, the ASR tools rank as follows: OpenAI, Speechmatics, RevAI, Amazon, and Microsoft.

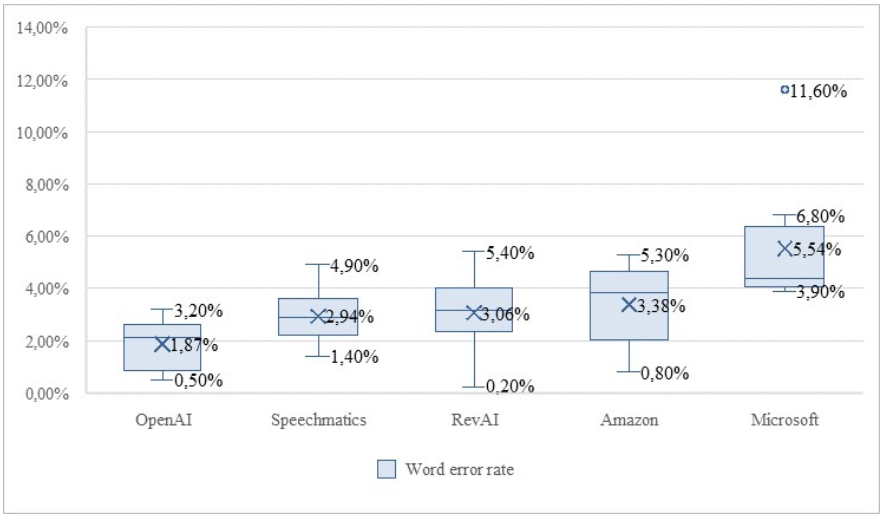


Fig. 2. ASR tool WER benchmarks in English

Figure 3 shows the WER for Latvian. All tools show a relatively similar range, except for Amazon's ASR tool, which exhibits a significantly higher average WER of 22,46%. This indicates that Amazon's accuracy in transcribing Latvian speech is considerably lower than the other tools. Based on Latvian language recognition, the ASR tools rank as follows: Speechmatics, Microsoft, RevAI, OpenAI, and Amazon.

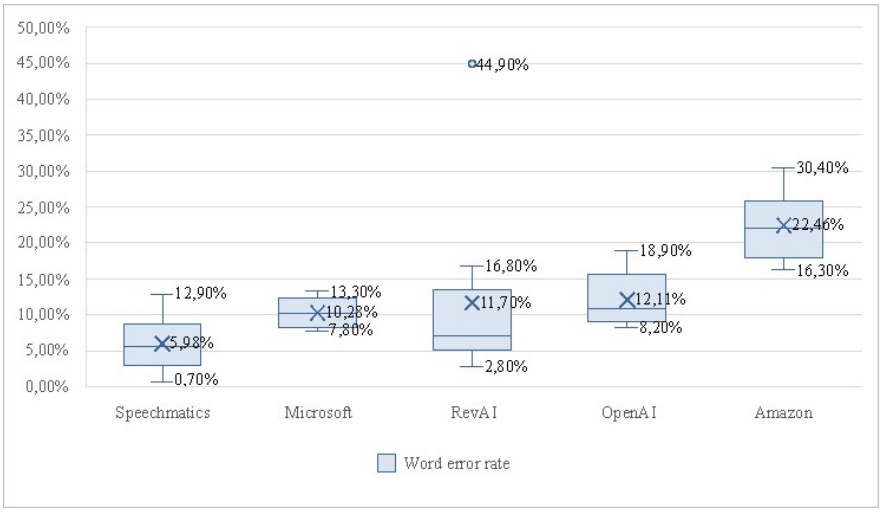


Fig. 3. ASR tool WER benchmarks in Latvian

Figure 4 shows the ASR tools ranked by their average WER in Russian. The ranking for Russian is: OpenAI, RevAI, Amazon, Speechmatics, and Microsoft. The ranking for Russian is: OpenAI, RevAI, Amazon, Speechmatics, and Microsoft.

Analysis of the WER across different target languages revealed that the tools' performance is relatively similar overall, with rankings varying depending on the language. However, Amazon's ASR tool stands out with its significantly higher WER for Latvian, indicating lower accuracy. Due to this lower performance, Amazon's ASR tool was excluded from further selection. The remaining tools proceeded to the next stage were: OpenAI, Speechmatics, RevAI, and Microsoft.

Stage 3: Benchmarking based on latency Processing latency measurements, using 15-second audio fragments, revealed substantial differences among the ASR tools:

- **OpenAI Whisper** demonstrated the lowest average latency at 1,53 seconds across all languages.
- **Microsoft Azure** had an average latency of 6,60 seconds.
- **RevAI** exhibited an average latency of 10,24 seconds.
- **Speechmatics** had the highest average latency at 13,99 seconds.

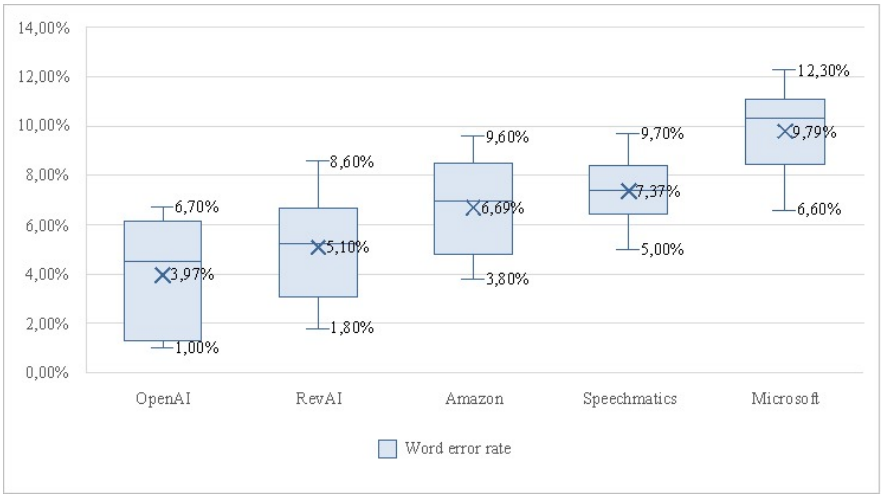


Fig. 4. ASR tool WER benchmarks in Russian

5.5 Conclusion

This study evaluated several ASR tools for real-time phone fraud detection based on transcription accuracy (WER) and latency. While most tools showed similar WER across languages, Amazon ASR underperformed significantly in Latvian. However, the key factor in the final selection was latency.

Based on the comprehensive evaluation, OpenAI Whisper was selected as the optimal transcription tool for integration into the phone fraud detection and prevention system. It offers several advantages:

- **High transcription accuracy** across all target languages, ensuring that the content and context of conversations are accurately captured.
- **Rapid processing speed,** with the lowest latency among the evaluated tools, enabling the system to detect and prevent fraudulent activities promptly.
- **Support for target languages,** including Latvian, English, and Russian, which are essential for the system's applicability in the regional context.
- **Automatic language detection,** allowing the tool to adapt to the language of any given conversation without manual intervention.

The selection of OpenAI Whisper aligns with the system's requirements for accuracy and speed, ensuring that the phone fraud detection and prevention system can operate effectively in real-time.

6 Content analysis tool selection

Selecting an appropriate large language model (LLM) is crucial for the success of a phone fraud detection and prevention system. This chapter presents an empirical study

to select the most suitable LLM for analyzing telephone conversations to detect and prevent phone fraud in real-time, necessitating both high accuracy and low latency. This study focuses on commercially available LLMs, assuming they benefit from greater resources for development and training data. Unlike automatic speech recognition (ASR) systems, where comparative evaluations are less common, LLMs have been extensively researched and compared, fueled by rapid advancements in artificial intelligence.

This study evaluates three leading commercial LLMs: Anthropic Claude, Google Gemini, and OpenAI GPT. These companies are recognized as industry leaders, actively comparing their models against each other (Achiam et al., 2023; Team et al., 2023; WEB, l). This study utilizes their most powerful models available at the time of research: Claude-Opus, Gemini 1.5-Pro, and GPT-4. The section describes the research methodology and results, culminating in the selection of the optimal LLM.

6.1 Methodology

This section details the methodology for selecting the LLM, including the prompts, dataset, and study limitations.

The study aims to:

1. Objectively evaluate the LLMs to select the most suitable one for real-time phone fraud detection and prevention.
2. Identify the optimal LLM prompt for this task.

LLM performance depends on both the model and its configuration. The optimal combination will ensure the best results in fraud detection and prevention.

The methodology evaluates three quantitative criteria:

- Response speed (latency): Measures the time taken by the LLM to process a text fragment and respond.
- Classification effectiveness: Evaluates the LLM's ability to detect fraudulent calls using standard classification metrics (accuracy, recall, precision, and F1 score). Effective detection enables timely prevention.
- Classification effectiveness variation: Evaluates the stability and consistency of LLM classification results across multiple runs, accounting for their generative nature and inherent randomness. To assess the variability and statistical significance of the results, the mean, standard deviation, and 95% confidence interval of the performance metrics will be calculated.

All three criteria—classification effectiveness, its consistency, and response speed—are critical for a real-time phone fraud detection and prevention system. Accurate and consistent fraud identification is essential to minimize financial loss and protect fraud victims, while a fast response time is necessary to prevent fraud from escalating. These criteria will be evaluated as a unit to determine the overall suitability of an LLM for real-time operation.

6.2 Study dataset

This study utilizes the same dataset (audio recordings) as the previous chapter on ASR tool selection (see section "5.2 Dataset preparation"). However, this study analyzes the transcribed texts from those recordings, generated by the best-performing ASR tool (OpenAI Whisper), as input for the LLMs. (For a full description of the dataset, see integrated system's evaluation section "7.1 Methodology", subsection "Study dataset").

Each scenario's text is divided into 15-second fragment. This division corresponds to the real-time system's operation, where conversations would be analyzed in 15-second intervals to provide enough context. The fragment length in words varies depending on the speech rate used during scenario generation for particular language. For example, in Latvian, with a speech rate of 170 words per minute, a 15-second fragment contains approximately 43 words.

To ensure anonymity and privacy during conversation content analysis, personally identifiable information (e.g., names and numbers) was removed from the transcribed texts and replaced with placeholders.

6.3 Latency evaluation methodology

This section describes the methodology for evaluating LLM latency, given the influence of prompts, LLM response speed is critical for real-time fraud detection. Latency was measured by sending transcribed conversation text fragments to each LLM via their public API and precisely timing the response. Different LLM and prompt combinations were tested to assess how prompt length and complexity affect response time. The results were analyzed, including classification performance and variation, to determine the optimal balance between speed and accuracy.

6.4 Classification effectiveness evaluation methodology

This section details the methodology for evaluating LLM classification effectiveness in phone fraud detection and prevention. The evaluation involves two stages: assessing detection effectiveness and assessing prevention potential by identifying fraud sufficiently early in the conversation.

Phone fraud detection evaluation

1. **LLM and prompt combinations:** Three LLMs (Claude-Opus, Gemini 1.5-Pro, GPT-4) and three different prompts focused on fraud detection were evaluated, analyzing all 9 possible combinations.
2. **Continuous classification:** Each scenario's text fragments were sequentially analyzed by the LLMs to detect fraud indicators. Analysis continued until the text ended or fraud was detected.
3. **Performance metric calculation:** Classification results were compared to manual annotations to calculate accuracy, recall, precision, and F1 score for each LLM and prompt combination.

4. **Performance variation calculation:** To mitigate randomness, each scenario was analyzed 5 times with each LLM and prompt combination. To obtain more deterministic results, the LLM temperature parameter was set to 0. The mean, standard deviation, and 95% confidence interval of performance metrics were calculated.
5. **Result analysis:** Performance metrics were compared across LLMs and prompt combinations to identify those with the best detection effectiveness.

Phone fraud prevention evaluation

1. **Continuous classification:** Similar to detection, each scenario's text fragments were sequentially analyzed by all LLM and prompt combinations.
2. **Fraud point determination:** When a fragment is classified as fraudulent, the LLM identifies the specific text portion triggering this classification. The position of the last word in this portion within the overall conversation is marked as the "fraud point." This point is then expressed as a percentage of the total conversation length ("fraud point ratio").
3. **Risk point determination:** "Risk point phrases" were manually annotated within each scenario. These phrases represent moments where the scammer could potentially obtain sensitive information or cause harm, regardless of the victim's response, e.g., a phrase "Please provide Your card number" is considered a risk point phrase. The "risk point" was identified as the last word of this phrase, and its position was calculated as a percentage of the total conversation length ("risk point ratio").
4. **Prevention potential assessment:** If the fraud point ratio was less than the risk point ratio, the LLM had successfully detected the scam before the critical risk point, indicating a successful prevention opportunity. This prevention rate was calculated for each LLM and prompt combination.
5. **Prevention potential variation:** Similar to detection evaluation, each scenario was analyzed 5 times with each LLM and prompt combination to mitigate randomness and calculate the prevention indicator variation.
6. **Result analysis:** The prevention rate was compared across LLMs and prompt combinations to identify those with the best prevention rate.

6.5 LLM prompts

LLM prompts are crucial for specifying the task, context, and desired output format. Effectively prompting the LLMs is essential for accurate and efficient fraud detection within the system. This study utilizes three distinct prompts to evaluate LLMs in the context of phone scam detection and prevention.

It's important to note that during the evaluation, the LLM continuously analyzes each conversation fragment, maintaining context and "remembering" the classification of previous fragments. This allows the model to build a comprehensive understanding of the conversation as it progresses, leading to more accurate assessments.

Prompt 1: Baseline prompt This prompt is simple and direct, asking the model to estimate the probability (as a percentage) that a conversation fragment is fraudulent without providing specific criteria. A 70% threshold is set for classifying a conversation as fraudulent. This prompt leverages "zero-shot prompting", assessing the model's inherent ability to detect fraud based on its existing knowledge. It also serves as a baseline for comparison with other prompts. The expected output is a JSON object with a percentage value representing the likelihood of fraud. Its concise nature and simple output format have the potential for the fastest response times.

Prompt 2: Fraud indicators and risk levels prompt. This prompt provides detailed information about fraud indicators and their risk levels, e.g., an unsolicited offer is a fraud indicator classified as medium risk. The model analyzes text fragments and accumulates information about detected indicators, such as emotional manipulation, impersonation of authority, urgency, and requests for sensitive information. This prompt utilizes a "few-shot prompting" technique to guide the models by providing specific fraud indicator examples. The expected output is a JSON object with the detected indicators and their risk levels. This prompt provides specific criteria for fraud detection, enabling more targeted analysis. Risk levels help the model assess the severity of detected indicators. However, this prompt is longer and requires more complex analysis and output, potentially increasing response time.

Prompt 3: Point system prompt. Similar to Prompt 2, but instead of risk levels, this prompt assigns points to each fraud indicator. The model accumulates points, and upon reaching a threshold, the conversation is flagged as fraudulent. The expected output is a JSON object with the accumulated points. This prompt evaluates the model's ability to assess the severity of indicators by assigning points. The point system allows for more flexible analysis, considering the cumulative effect of indicators. This prompt also requires complex analysis and output. However, it may be faster than Prompt 2 due to the absence of logical checks for different risk level thresholds.

6.6 Limitations and assumptions

The study acknowledged several limitations and assumptions. The predefined fraud indicators in Prompts 2 and 3 may limit the LLM's ability to detect novel or subtle fraud tactics. LLMs are still under development and may make errors or misinterpret context, potentially affecting their accuracy. The study used a controlled environment with transcribed scenarios, isolating LLM performance from real-world complexities like noisy audio, interruptions, and overlapping speech.

The study also relied on several assumptions. Latency measurements assumed a stable network connection, which may not always be the case in real-world applications. The study assumed high accuracy of transcriptions obtained using OpenAI Whisper, although transcription errors can occur and may affect the LLM's analysis.

6.7 Results

This section summarizes the results, addressing the objectives outlined in the methodology: identifying the most suitable conversation content analysis tool and the optimal prompt, considering classification effectiveness, its variation, and latency.

A total of 1350 individual classifications were performed for the comparative evaluation of classification effectiveness: 10 phone conversation scenarios × 5 iterations × 3 languages × 3 models × 3 prompts. The total number of measurements is 7937 (the number of text fragments analyzed to achieve individual classifications and latency evaluation).

6.7.1 Comparative evaluation of classification effectiveness This section presents the results of the comparative evaluation of the classification effectiveness of the conversation content analysis tools, focusing on three main indicators: accuracy, F1 score, and prevention rate. These indicators together provide a comprehensive picture of the ability of LLM models to effectively detect and prevent phone fraud.

Phone fraud detection This section evaluates the ability of LLMs to detect phone fraud by classifying phone conversation scenarios as fraudulent or legitimate. We use two key metrics: accuracy, reflecting the model's ability to correctly classify both fraudulent and legitimate conversations, and the F1 score, which combines precision and recall to provide a balanced evaluation of the model's performance in classifying fraudulent calls.

Figure 5 displays the average accuracy and F1 scores for each LLM across all prompts and target languages, including the 95% confidence intervals derived from variations in classification results for each prompt across languages.

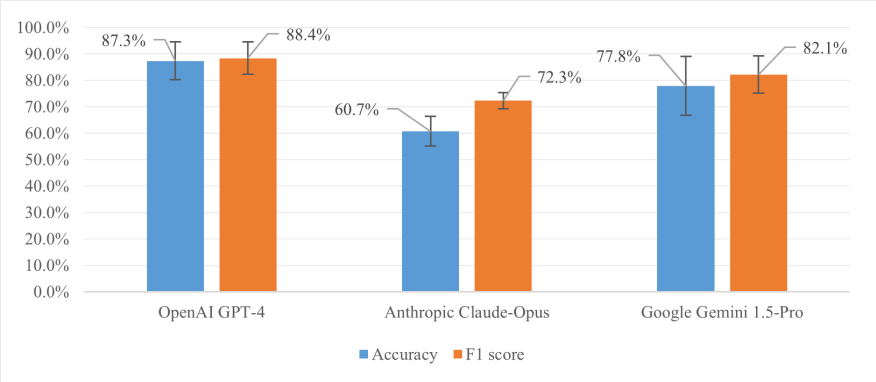


Fig. 5. LLM classification performance at 95% confidence interval

OpenAI GPT-4 demonstrates the best average performance with the highest accuracy and F1 scores, indicating its strong ability to accurately classify phone conversa-

tions. Google Gemini 1.5-Pro follows closely with slightly lower but still strong results, demonstrating its effectiveness in fraud detection. Anthropic Claude-Opus shows the weakest performance, exhibiting lower accuracy and F1 scores, which suggests potential difficulties in accurately distinguishing fraudulent calls from legitimate ones.

This lower performance may stem from Anthropic's cautious approach and its tendency to classify conversations as fraudulent based on limited context, such as the mere mention of a scam attempt, regardless of the overall conversation's nature. This behavior results in an increased number of false positives, misclassifying legitimate conversations as fraudulent.

Figure 6 analyzes the impact of each prompt on classification effectiveness, showing the average accuracy and F1 scores for all LLMs, including 95% confidence intervals derived from variations in classification for each LLM across languages.

Key observations from this analysis include:

- Prompt No. 1, leveraging the models' internal understanding and a "zero-shot" learning approach, yields the highest average accuracy and F1 scores. This suggests that allowing LLMs to interpret fraud indicators independently, without explicit criteria, can be more effective in detecting complex scam tactics.
- Prompt No. 2, based on predefined fraud characteristics and risk levels, shows balanced performance with the least variation across LLMs. While providing more consistent results, this approach may lead to overly literal interpretations and false positives when nuanced context is crucial.
- Prompt No. 3, utilizing a point system for fraud indicators, produces the lowest average accuracy and F1 scores. This may be attributed to challenges in accurately assigning points to indicators and defining an appropriate threshold for fraud detection.

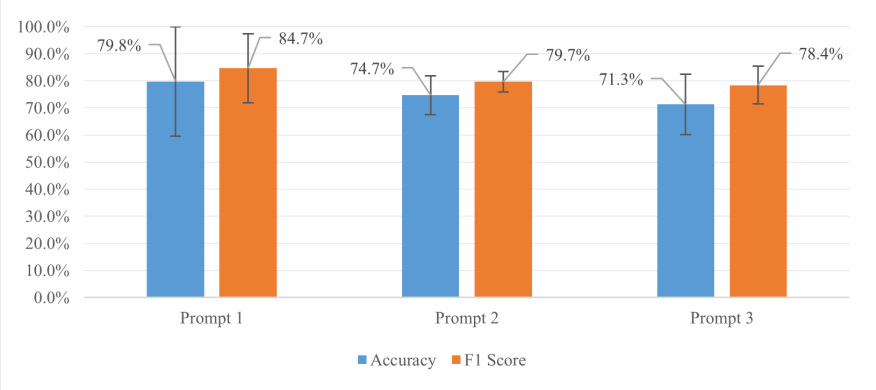


Fig. 6. Impact of prompts on classification performance at 95% confidence interval, aggregating all LLM results

Table 1 provides a more detailed comparison of classification performance across different LLM and prompt combinations, showing average accuracy and F1 scores with 95% confidence intervals derived from variations across languages.

Table 1. Comparative evaluation of LLMs by classification performance metrics

LLM	Prompt	Accuracy (avg.)	95% CI (accuracy)	F1 (avg.)	95% CI (F1)
OpenAI GPT-4	Prompt 1	96,0%	91,4% - 100,0%	95,7%	90,5% - 100,0%
OpenAI GPT-4	Prompt 2	80,0%	70,7% - 89,3%	81,8%	74,2% - 89,4%
OpenAI GPT-4	Prompt 3	86,0%	78,4% - 93,6%	87,6%	81,1% - 94,1%
Google Gemini 1.5-Pro	Prompt 1	90,0%	88,2% - 91,8%	90,0%	88,6% - 91,4%
Google Gemini 1.5-Pro	Prompt 2	78,7%	67,3% - 90,1%	82,6%	73,9% - 91,3%
Google Gemini 1.5-Pro	Prompt 3	64,7%	55,0% - 74,3%	73,8%	68,9% - 78,6%
Anthropic Claude-Opus	Prompt 1	53,3%	48,3% - 58,4%	68,3%	65,8% - 70,7%
Anthropic Claude-Opus	Prompt 2	65,3%	54,1% - 76,6%	74,8%	68,2% - 81,3%
Anthropic Claude-Opus	Prompt 3	63,3%	49,9% - 76,7%	73,8%	66,3% - 81,3%

Key findings from this table include:

- The superiority of Prompt No. 1 in enabling both OpenAI and Google models to achieve their highest accuracy and F1 scores. This emphasizes the effectiveness of leveraging the models' inherent knowledge and understanding of fraud.
- The trade-off between OpenAI and Google: While OpenAI GPT-4 with Prompt No. 1 shows slightly higher average scores, its wider confidence intervals indicate greater performance variation across languages and iterations, raising concerns about consistency. Google Gemini 1.5-Pro with Prompt No. 1 demonstrates more stable and predictable performance, albeit with slightly lower average scores.
- The limitations of Anthropic Claude-Opus: Despite achieving perfect recall (detecting all fraudulent calls) across all prompts (as shown in Table 2), Anthropic consistently exhibits the lowest accuracy and F1 scores due to its significantly lower precision and high false positive rate.

Table 2. Anthropic Claude-Opus classification performance metrics: precision and recall

LLM	Prompt	Precision (avg.)	95% CI (precision)	Recall (avg.)	95% CI (recall)
Anthropic Claude-Opus	Prompt 1	51,9%	49,0% - 54,7%	100,0%	100,0% - 100,0%
Anthropic Claude-Opus	Prompt 2	60,1%	51,5% - 68,8%	100,0%	100,0% - 100,0%
Anthropic Claude-Opus	Prompt 3	59,0%	49,2% - 68,7%	100,0%	100,0% - 100,0%

These findings highlight that while Anthropic can effectively identify all fraudulent calls, its high false positive rate makes it less suitable for a real-time system where minimizing disruption to legitimate conversations is crucial.

Phone fraud prevention This section analyzes the ability of LLMs to proactively prevent phone fraud by identifying fraudulent activity early in a conversation. We focus on the prevention rate, a metric reflecting the model's ability to detect fraud before it reaches a critical "risk point" where the scammer is likely to obtain sensitive information or money.

Figure 7 shows the average prevention rates for each LLM across all prompts and languages, including the 95% confidence interval derived from variations between prevention rates for each prompt across languages.

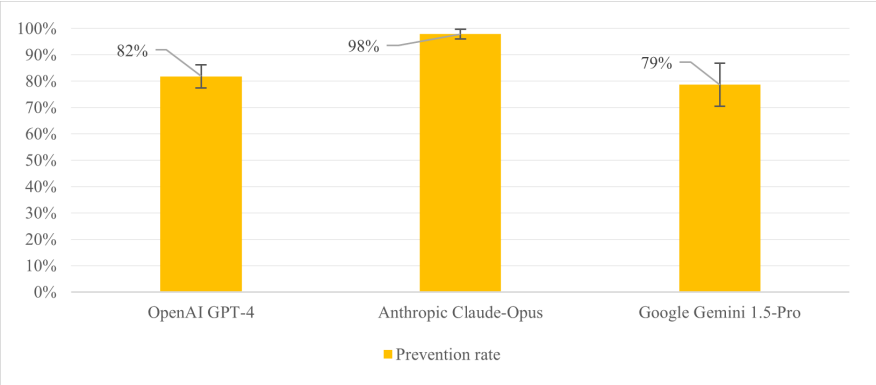


Fig. 7. Average LLM prevention rate at 95% confidence interval, aggregating all LLM results

Anthropic Claude-Opus demonstrates the highest average prevention rate (98%) with minimal variation, indicating its strong ability to detect scams early and prevent potential harm. Google Gemini 1.5-Pro and OpenAI GPT-4 also perform well, achieving prevention rates of 79% and 82% respectively, demonstrating their effectiveness in fraud prevention.

Figure 8 displays the average prevention rates for each prompt, taking into account the results from all LLMs. The 95% confidence intervals reflect the variation in prevention rates between different LLMs for each prompt across languages.

Prompt No. 1, which leverages the models' inherent understanding, achieves the highest average prevention rate (91,1%). This suggests that this prompting strategy allows the models to effectively identify fraudulent behavior before the risk point is reached. Prompt No. 3, based on a feature point system for fraud detection, also shows strong performance with an average prevention rate of 85,8%. In contrast, Prompt No. 2, which focuses on explicit fraud characteristics and risk levels, yields the lowest average prevention rate (81,3%). All prompts exhibit moderate variation in their prevention rates across the different LLMs.

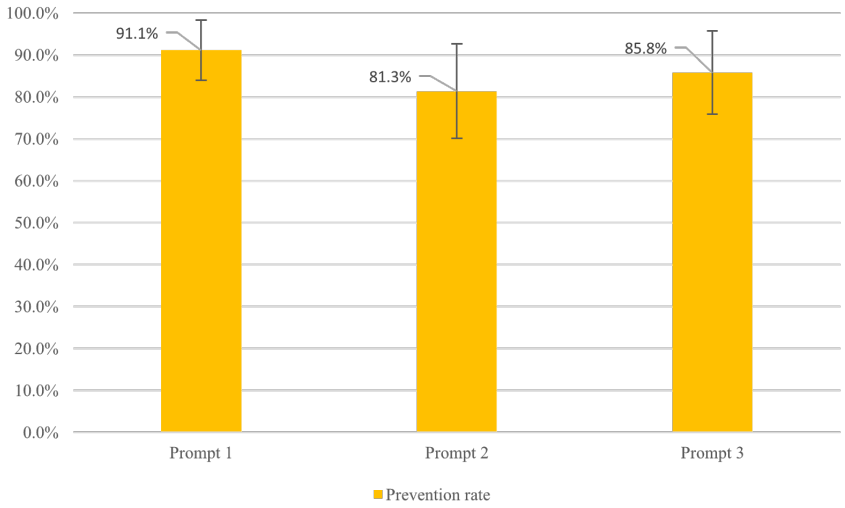


Fig. 8. Average prevention rate of prompts at 95% confidence interval, aggregating all LLM results

Table 3 provides a more granular view, presenting the average prevention rates for each specific LLM and prompt combination, along with their 95% confidence intervals derived from variations across languages.

Key observations from this analysis include:

- Anthropic's consistent excellence in prevention across all prompts, achieving a perfect 100% prevention rate with Prompt No. 1. This highlights the model's exceptional capability in early fraud detection.
- The superior effectiveness of Prompt No. 1 in eliciting high prevention rates from all LLMs. This underscores the value of leveraging the models' inherent understanding and employing a "zero-shot" prompting approach.
- Google Gemini 1.5-Pro's reliability in prevention when used with Prompt No. 1. This combination not only achieves a high prevention rate (89,3%) but also exhibits a narrow confidence interval (87,3% - 91,4%), indicating consistent and stable performance. In comparison, OpenAI GPT-4 with the same prompt shows a slightly lower prevention rate (84%) and wider variation (77,9% - 90,1%).

Despite Anthropic's strong performance in prevention, concerns regarding its accuracy and tendency towards false positives, as detailed in the previous section, make it unsuitable for a real-time system.

Considering both detection (classification effectiveness) and prevention capabilities, Google Gemini 1.5-Pro with Prompt No. 1 emerges as a compelling candidate for a real-time fraud prevention system. This combination offers a balance of accurate fraud detection and minimal disruption to legitimate conversations. However, the final model selection will also consider latency performance, which is analyzed in the next section.

Table 3. Comparative evaluation of LLMs by prevention rate

LLM	Prompt	Prevention rate (avg.)	95% CI
OpenAI GPT-4	Prompt 1	84,0%	77,9% - 90,1%
OpenAI GPT-4	Prompt 2	76,0%	69,0% - 83,0%
OpenAI GPT-4	Prompt 3	85,3%	74,1% - 96,6%
Google Gemini 1.5-Pro	Prompt 1	89,3%	87,3% - 91,4%
Google Gemini 1.5-Pro	Prompt 2	72,0%	62,7% - 81,3%
Google Gemini 1.5-Pro	Prompt 3	74,7%	67,4% - 82,0%
Anthropic Claude-Opus	Prompt 1	100,0%	100,0% - 100,0%
Anthropic Claude-Opus	Prompt 2	96,0%	89,9% - 100,0%
Anthropic Claude-Opus	Prompt 3	97,3%	93,3% - 100,0%

6.7.2 Comparative latency evaluation While classification effectiveness and its variation are primary considerations when selecting an LLM for a real-time phone fraud detection and prevention system, latency, or response speed, is also crucial. This section analyzes the latency of different LLM and prompt combinations to assess their suitability for real-time analysis.

Table 4 shows the average latency for each prompt, aggregating the results of all LLMs. Prompt No. 1 exhibits the lowest average latency (3,33 seconds), followed by Prompt No. 3 (3,51 seconds) and Prompt No. 2 (4,63 seconds). These results confirm the prediction in the LLM prompt section regarding the impact of prompt length, complexity, and output format on response time.

Table 4. Average latency per prompt

Prompt	Average Latency (seconds)
Prompt 1	3,33
Prompt 2	4,63
Prompt 3	3,51

Table 5 presents the latency analysis of the LLMs. Anthropic Claude-Opus demonstrates the slowest response time (4,57 seconds), further supporting previous conclusions about its unsuitability for a real-time system. Google Gemini 1.5-Pro stands out with the lowest average latency (2,69 seconds), indicating its ability to quickly process information and provide analysis responses. OpenAI GPT-4 takes second place with an average latency of 4,20 seconds.

As previously mentioned, choosing between OpenAI GPT-4 and Google Gemini 1.5-Pro with Prompt No. 1 presents a trade-off. While GPT-4 showed better classification effectiveness, its wider confidence intervals suggest greater variation and po-

Table 5. Average latency per LLM

LLM	Average Latency (seconds)
Anthropic Claude-Opus	4,57
Google Gemini 1.5-Pro	2,69
OpenAI GPT-4	4,20

tentially lower consistency compared to Gemini 1.5-Pro. Gemini, on the other hand, demonstrated slightly lower classification scores but with narrower confidence intervals, indicating more stable and reliable performance. Additionally, Gemini 1.5-Pro exhibited a better prevention rate with a narrower confidence interval than GPT-4.

To make a final decision, it's necessary to evaluate the variation in latency and its impact on real-time system operation. Figures 9 and 10 illustrate the latency distribution for OpenAI GPT-4 and Google Gemini 1.5-Pro, respectively, allowing us to assess the stability and consistency of their response times, which is critical for a real-time system aimed at timely phone fraud detection.

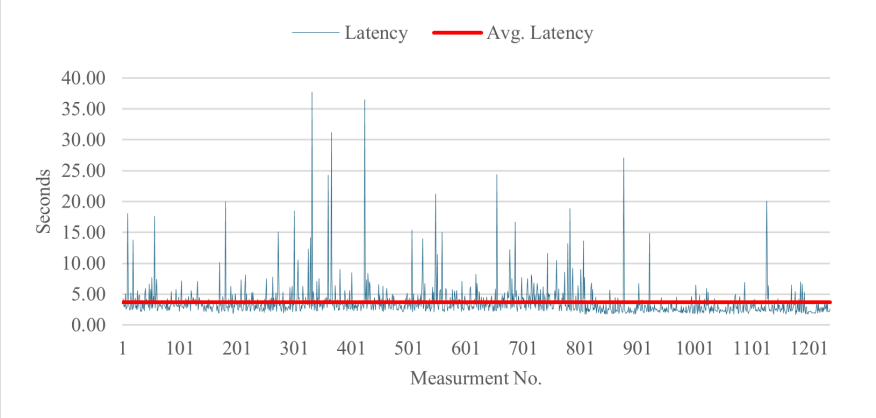


Fig. 9. Variation of OpenAI GPT-4 latency measurements with Prompt No. 1

The data reveals the following:

- OpenAI model's latency instability: Figure 9 shows that while OpenAI GPT-4 has an average latency of 3,66 seconds with Prompt No. 1, the latency measurements are highly variable, with some exceeding 10 seconds and even reaching almost 40 seconds. This instability raises concerns about the model's reliability in a real-time system.
- Google model's latency stability: In contrast, Figure 10 shows that Google Gemini 1.5-Pro has a more stable and consistent latency, mostly remaining within the

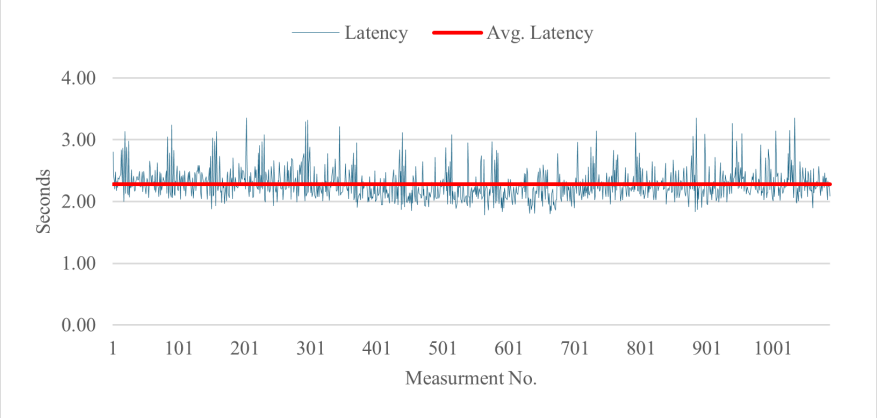


Fig. 10. Variation of Google Gemini 1.5-Pro latency measurements with Prompt No. 1

range of 1,75 to 3,4 seconds, with average latency being 2,28 seconds. This stability ensures a predictable and reliable response time, crucial for real-time system operation.

6.8 Conclusions

This evaluation identified Google Gemini 1.5-Pro with Prompt No. 1 as the optimal choice for a real-time phone fraud detection and prevention system. This conclusion is based on its superior balance of classification effectiveness, prevention rate, and latency, with a strong emphasis on consistency and real-time performance.

While OpenAI GPT-4 initially showed promising accuracy (96%) and F1 score (95,7%), its wider confidence intervals revealed significant performance variation across languages and iterations, raising concerns about its reliability in a real-time setting. Furthermore, its lower prevention rate (84%) compared to Google further supports this decision.

Google Gemini 1.5-Pro consistently demonstrated a strong balance between accuracy (90%) and F1 score (90%) across all languages with narrower confidence intervals, indicating greater stability and predictability. This ensures reliable fraud detection while minimizing disruptive false positives. Moreover, its impressive prevention rate (89,3%) and consistently low latency (averaging 2,28 seconds) solidify its suitability for real-time fraud detection and prevention.

Anthropic Claude-Opus, despite its high prevention rate, proved unsuitable due to lower classification accuracy, a high false positive rate, and the highest average latency among the models tested.

Ultimately, Google Gemini 1.5-Pro with Prompt No. 1 best meet's the system requirements and priorities. Its balanced performance, low latency, and consistent reliability make it an effective and practical solution for real-time phone fraud detection and prevention.

7 System evaluation

This chapter presents an empirical study evaluating the effectiveness of a phone fraud detection and prevention system. The system integrated a pre-selected transcription tool (OpenAI Whisper) and a large language model (Google Gemini 1.5-Pro with Prompt No. 1) to analyze phone conversations in real-time. The study employed an expanded dataset, and a modified methodology tailored to the real-time requirements of the integrated system. The primary goals were to:

- Assess the system's accuracy in classifying phone calls as fraudulent or legitimate.
- Evaluate its ability to prevent fraud through early detection.
- Analyze the system's latency and suitability for real-time operation.

7.1 Methodology

This section details the methodology for evaluating the integrated system. The goal was to objectively assess its performance across various metrics (prevention, accuracy, F1 score, precision, and recall).

The methodology was adapted from previous chapters ("5. Transcription tool selection" and "6. Content analysis tool selection") to accommodate the real-time requirements of the integrated system.

Study dataset. The study employed an expanded dataset, comprising 30 phone conversation scenarios (15 fraudulent, 15 legitimate) translated into Latvian, English, and Russian, resulting in a total of 90 scenarios.

Fraudulent scenarios were sourced from YouTube recordings, reflecting common scam types prevalent in Latvia. Legitimate scenarios, developed in consultation with a senior security expert from one of the largest banks in Latvia, simulate real banking practices and communication styles and represented a control group. This balanced dataset enabled objective evaluation of the system's performance in detecting and preventing diverse scam tactics.

Use of computer-generated voice recordings To ensure consistent testing conditions and control speech rate, the study employed computer-generated voice recordings with specific words per minute (WPM) for each language: English (200 WPM), Latvian (170 WPM), and Russian (160 WPM). These WPM values, based on typical speech rates and adjusted for potential stress-induced acceleration (Rodero, 2012; Kappen et al., 2024; Bogdanovs, 2018), enabled precise calculation of conversation progression for real-time analysis.

Within the simulated real-time environment, this controlled speech rate allowed for precise calculation of conversation progression, enabling accurate assessment of the system's ability to detect and prevent scams before critical points are reached. Further details regarding the methodology and the impact of WPM are provided in subsequent section "Classification effectiveness evaluation."

Latency analysis The latency evaluation methodology at the system level remained similar to the previously described LLM latency assessment. However, in this case, the total time required for the system to process 15-second text fragments and provide a response was measured. This included:

1. **Audio transcription:** Time taken by OpenAI Whisper to transcribe the audio.
2. **Text processing:** Time required for creating the audio recording (simulated by adding 0,40 seconds, the maximum time needed for combining audio tracks in the real-time system) and automatically removing personally identifiable information from the transcribed text.
3. **LLM analysis:** Time taken by Google Gemini 1.5-Pro to process the edited text fragment and provide a response.

These three stages constituted the system's total latency, measured for each text fragment. The results were analyzed to evaluate the system's suitability for real-time operation and to calculate the "real-time word" position for assessing prevention effectiveness (explained in the subsequent section).

Classification effectiveness evaluation The methodology for evaluating classification effectiveness largely followed the principles used for individual LLM assessment in the previous chapter, with modifications to reflect the integrated system evaluation and adapt to real-time analysis challenges.

Similar to the previous chapter, the evaluation included continuous classification, performance metric calculation, variation analysis, and result analysis using the full dataset in 3 target languages. However, to accurately assess fraud prevention effectiveness in a simulated real-time environment, a modified fraud point calculation methodology was introduced, incorporating the time factor.

Fraud point calculation – adapting to real-time environment The simulated environment presented challenges in accurately assessing real-time fraud prevention effectiveness. In a real-world setting, prevention is determined by whether a call is classified and terminated before reaching the "risk point". In the simulated environment, however, the system analyzes segments sequentially, potentially missing the risk point as the conversation progresses in real-time.

To address this, a modified fraud point calculation methodology was implemented, incorporating the time factor (illustrated in Figure 11):

1. **Continuous classification:** Each 15-second text fragment was sequentially analyzed using Google Gemini 1.5-Pro with Prompt No. 1 until fraud was detected or the conversation ended.
2. **Last word identification:** Upon fraud detection, the last word in the transcribed and edited 15-second fragment was identified, representing the point reached in the real-time conversation.
3. **Processing time measurement:** The total time taken by the system to process the fragment, from transcription to analysis results, was measured. This included transcription, text editing, and LLM analysis.

4. **Additional spoken words calculation:** Using the respective language's WPM, the average number of words spoken per second was calculated and multiplied by the processing time to determine the number of words spoken during analysis.
5. **"Real-time word" position determination:** The additional spoken words were added to the last word's position (from step 2) to determine the "real-time word" position – the point the conversation would have reached after analysis (see Figure 11).
6. **Fraud point calculation:** The "real-time word" position was used to calculate the fraud point ratio relative to the total words in the scenario.
7. **Comparison with risk point:** The fraud point ratio was compared to the risk point ratio to determine if the system detected fraud before the critical risk point, considering processing time.

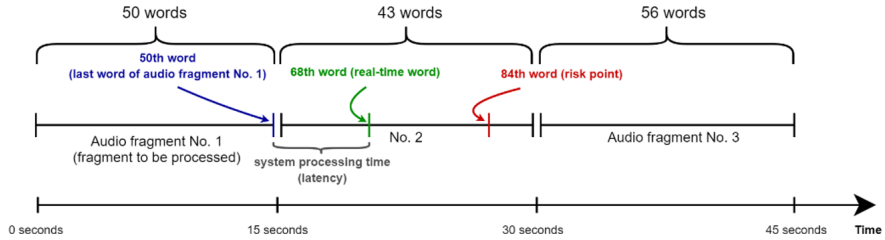


Fig. 11. Illustration of the customized phone fraud prevention methodology

This methodology ensured an objective system evaluation simulating real-time conditions, enabling conclusions about real-time phone scam detection and prevention.

7.2 Limitations and assumptions

While the system evaluation provides valuable insights, it is essential to acknowledge limitations and assumptions. The evaluation assumed a stable network connection for uninterrupted operation of both the transcription tool and the LLM API. Real-world network instability could impact system performance and latency. Although expanded, the dataset may not encompass all possible phone scam scenarios and tactics. While offering controlled conditions, the simulated environment cannot fully replicate the complexity and unpredictability of a real-time system.

7.3 Results

This section presents the results of the system evaluation, addressing the objectives outlined in the methodology section. The evaluation involved 450 individual classifications (30 scenarios x 5 iterations x 3 languages), totaling 2816 measurements (analyzed text fragments for classification and latency assessment).

7.3.1 Latency System latency is crucial for real-time fraud prevention. On average, audio transcription took 1,82 seconds, text processing (including audio track combining simulation and text editing) took 0,43 seconds, and LLM analysis took 2,26 seconds, constituting the largest portion of the total system latency (4,52 seconds).

Minor discrepancies were observed compared to previous latency measurements in transcription tool evaluation chapter. Transcription latency was slightly higher (1,82 seconds vs. 1,53 seconds), potentially due to increased load on OpenAI servers. However, the LLM analysis latency (2,26 seconds) remained consistent with previous findings (2,28 seconds).

7.3.2 Classification effectiveness This section analyzes the integrated system's effectiveness in detecting and preventing phone scams using the expanded dataset and modified fraud point calculation. Four metrics were used to assess detection performance: accuracy, F1 score, precision, and recall. Prevention performance was evaluated using the prevention rate.

Phone scam detection Figure 12 presents the average values for each metric across all languages and iterations, including 95% confidence intervals derived from variations across target languages.

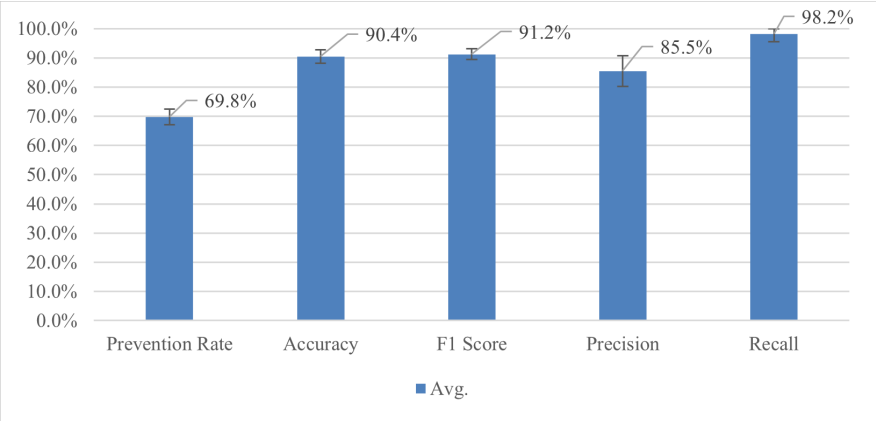


Fig. 12. System's average performance metrics at 95% confidence interval

The data indicates:

- **High classification effectiveness:** The system achieved an average accuracy of 90,4% and an F1 score of 91,2%, demonstrating its ability to accurately distinguish fraudulent and legitimate calls.
- **Excellent recall:** The system exhibited a recall of 98,2%, indicating its ability to detect almost all actual scam cases.

- **High precision:** The system's precision was 85,5%, suggesting a relatively low rate of false positives.
- **Moderate prevention rate:** The system achieved a prevention rate of 69,8%, successfully preventing scams in almost 70% of cases.

The narrow confidence intervals suggest consistent classification performance across scenarios and languages, crucial for reliable real-time operation.

Comparing the integrated system's performance to the isolated Google Gemini 1.5-Pro results (from "6. Content analysis tool selection"), the classification effectiveness was similar. The integrated system's average accuracy (90,4%) and F1 score (91,2%) aligned with the previous findings (90% accuracy and 90% F1 score). Similarly, precision (85,5%) and recall (98,2%) were comparable to the isolated model's performance (90,4% precision and 90,7% recall).

Despite minor variations, the integrated system's average values for three out of four classification metrics fell within the confidence intervals of the isolated Google model with Prompt No. 1:

- **Accuracy:** 90,4% (previous CI: 88,2% - 91,8%)
- **F1 score:** 91,2% (previous CI: 88,6% - 91,4%)
- **Precision:** 85,5% (previous CI: 85,6% - 95,3%) - only slightly below the lower bound due to rounding.
- **Recall:** 98,2% (previous CI: 88,6% - 92,7%) - exceeding the upper bound, likely attributed to the expanded dataset providing more examples for fraud identification.

This demonstrates that the integrated system achieved comparable classification effectiveness with an expanded dataset, while also providing real-time transcription and scam detection.

Phone scam prevention The integrated system's average prevention rate (69,8%, 95% CI: 67,1% - 72,5%) was lower than the isolated Google model's (89,3%, 95% CI: 87,3% - 91,4%). This difference can be primarily attributed to the use of an expanded dataset that incorporates a broader variety of real-world scenarios, as well as the inclusion of real-time factors (such as processing time and the "real-time word" position) in the fraud point calculation.

To assess the impact of latency, a "potential prevention rate" (ignoring processing time) was calculated (76%). This minor difference (6,2%) indicates that latency had a limited effect. The primary factor for the lower rate was the broader variety of scenarios present in the expanded dataset.

7.4 Conclusions

The integrated system demonstrated strong performance in both scam detection and prevention, achieving high accuracy (90,4%), F1 score (91,2%), and recall (98,2%). While the prevention rate (69,8%) was lower than the isolated LLM evaluation, it still highlights the system's potential for real-time fraud prevention.

The system's classification effectiveness was comparable to the isolated Google Gemini 1.5-Pro model, confirming its ability to maintain high accuracy and reliability with a larger dataset while providing real-time transcription and analysis.

Narrow confidence intervals across metrics indicate system stability and consistency across languages and iterations. The lower prevention rate is primarily attributed to the expanded dataset's complexity and minimally to the modified fraud point calculation for real-time simulation.

The system's average latency was 4,52 seconds, with 1,82 seconds for transcription, 0,43 seconds for text processing, and 2,26 seconds for LLM analysis. This latency is acceptable for real-time fraud detection, allowing timely analysis and identification of fraudulent indicators and thus timely prevention.

8 Discussion

In this study, we presented an integrated system for phone fraud detection and prevention that leverages AI tools to analyze phone conversation content in a simulated real-time environment. The evaluation was conducted using a dataset of fraudulent and legitimate transcripts derived from real-world call scenarios, with synthetic audio recordings. Our approach not only assesses fraud detection performance using key metrics such as accuracy, F1 score, precision, and recall but also emphasizes the system's capability to prevent fraud—an aspect that is often overlooked in the literature. This dual focus on both detection and prevention provides a comprehensive evaluation of the system's potential for practical application. While the results are promising in this controlled environment, they should be interpreted with the understanding that the evaluation setting may differ from live real-world conditions.

8.1 System advantages

The developed prototype for phone scam detection and prevention offers several advantages. By integrating AI-based tools for speech recognition and natural language processing, the system effectively complements traditional call filtering methods. Unlike simpler approaches, the system performs in-depth analysis of conversation content, enabling the identification of complex scam tactics and social engineering techniques. Furthermore, its language-agnostic design facilitates multilingual scam detection. The system's flexible architecture, based on microservices, allows for easy updates and future integration of improved AI models.

8.2 Limitations and criticism

Despite promising results, the system has limitations. Data privacy is a key concern, requiring strict adherence to data protection regulations. Automated data editing may lead to errors and potential information leakage, necessitating robust quality control. The dataset, while expanded, remains limited in its representation of potential scam scenarios and linguistic variations. Additionally, the evaluation was conducted using synthetic audio rather than real phone call recordings. While the synthetic audio was designed to reflect real-world scenarios and replicate telecom-grade quality, it may not fully capture the complexities of live phone conversation audio, including background noise, speaker variability, and mobile network disruptions.

The system's dependence on automatic speech recognition is a critical factor. Inaccuracies in transcription and audio segmentation can distort conversational context and hinder accurate classification. In real-world settings, challenges such as noisy environments, mobile network interruptions and linguistic nuances may exacerbate ASR errors—potentially leading to significantly higher word error rates that could compromise overall system effectiveness.

Finally, The chosen prompt may not be universally optimal, and the inherent variability in LLM outputs can influence system reliability.

9 Conclusions and future work

This study addressed the pressing issue of phone fraud and scams, which cause significant financial losses and emotional distress to telecommunication users globally.

Traditional detection methods, such as blacklists and CDR analysis, are becoming less effective due to scammers' evolving tactics (e.g., caller ID spoofing). This necessitates innovative approaches to combat phone fraud.

Conversation content analysis has emerged as a promising avenue for real-time scam detection and prevention. This approach leverages advancements in artificial intelligence, specifically automatic speech recognition (ASR) and large language models (LLM), to analyze the content of phone conversations and identify fraudulent patterns.

This study developed a prototype system integrating two AI tools: OpenAI Whisper for ASR and Google Gemini 1.5-Pro for LLM-based content analysis. Each tool was selected through a rigorous evaluation methodology.

Empirical evaluation of the integrated system in a simulated real-time environment demonstrated its high classification effectiveness. The system achieved strong performance across key metrics and the results indicate the system's ability to effectively detect fraud while maintaining a low false positive rate.

Furthermore, the system achieved a prevention rate of 69,8% (95% CI: 67,1% - 72,5%), demonstrating its potential for real-time intervention and prevention of phone fraud. While much research focuses on detection, this study highlights the importance of evaluating and optimizing systems for proactive prevention, aiming to stop scams before they cause harm. It is crucial to recognize that mere identification of a fraud attempt is not equivalent to prevention. A system might accurately identify a fraudulent call, but if the identification occurs too late in the conversation, it may not be possible to prevent negative consequences.

The system offers several advantages that make it a promising solution for combating phone fraud. It can be integrated with existing protection methods, such as blacklists and call filtering, creating a multi-layered defense system. The system's ability to analyze conversation content in real-time allows for rapid response to scam attempts. Multilingual support broadens its applicability across different regions. Finally, the flexible microservices architecture enables easy adaptation and updates, allowing for the integration of newer and improved AI tools as needed.

However, the evaluation also revealed limitations that need to be acknowledged. Data privacy remains a critical concern, requiring careful consideration and robust security measures. The potential for data editing errors and unintended information leakage

necessitates further research and development of more reliable techniques. The limited size and diversity of the dataset may affect the system's generalizability. Furthermore, challenges remain in optimizing prompts for specific scam types and managing the inherent variability in LLM outputs. Finally, errors in speech recognition and audio segmentation can lead to contextual distortions that hinder system effectiveness.

Future research should focus on addressing current limitations. This includes conducting real-world testing using actual phone call audio to evaluate system performance under real conditions and expanding the dataset to encompass a wider range of scenarios, languages and linguistic variations to assess generalizability. The use of synthetic audio recordings—while designed to replicate real-world conditions—may not fully capture the complexities of live phone conversations, such as background noise, speaker variability, and mobile network disruptions. Furthermore, our evaluation used a balanced dataset for controlled assessment. However, in real-world scenarios, fraudulent calls are far less frequent, which may reduce precision as false positives could outweigh true fraud cases. Future work should focus on adapting the system to this imbalance for more realistic performance. Additionally, exploring the integration of streaming ASR solutions and evaluating alternative file formats (e.g., lossless FLAC) for transcription could help reduce latency and improve accuracy. Investigating separate speaker transcription and timestamped utterances could enhance fraud detection by providing clearer conversational context for the LLM. Further investigation of prompt engineering and optimization techniques is needed. Exploring strategies to mitigate variability in LLM outputsand evaluating the feasibility of open-source AI tools could lead to more robust and cost-effective solutions. Finally, expanding the application of LLMs to detect fraud in other communication channels, such as SMS and email, could contribute to a more comprehensive approach to fraud prevention.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023). Gpt-4 technical report, *arXiv preprint arXiv:2303.08774* .

Bogdanovs, M. (2018). Runas ātrums audiovizuālajos materiālos un tā ietekme uz audiovizuālās tulkošanas stratēģijām [Speech rate in audiovisual materials and its impact on audiovisual translation strategies].

Derakhshan, A., Harris, I. G., Behzadi, M. (2021). Detecting telephone-based social engineering attacks using scam signatures, *Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics*, pp. 67–73.

Ferraro, A., Galli, A., La Gatta, V., Postiglione, M. (2023). Benchmarking open source and paid services for speech to text: an analysis of quality and input variety, *Frontiers in big Data* **6**, 1210559.

Filippidou, F., Moussiades, L. (2020). A benchmarking of ibm, google and wit automatic speech recognition systems, *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I 16*, Springer, pp. 73–82.

Gowri, S. M., Sharang Ramana, G., Sree Ranjani, M., Tharani, T. (2021). Detection of telephony spam and scams using recurrent neural network (rnn) algorithm, *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1, pp. 1284–1288.

Hong, B., Connie, T., Goh, M. K. O. (2023). Scam calls detection using machine learning approaches, *2023 11th International Conference on Information and Communication Technology (ICoICT)*, IEEE, pp. 442–447.

Kappen, M., Vanhollebeke, G., Van Der Donckt, J., Van Hoecke, S., Vanderhasselt, M.-A. (2024). Acoustic and prosodic speech features reflect physiological stress but not isolated negative affect: a multi-paradigm study on psychosocial stressors, *Scientific Reports* **14**(1), 5515.

Li, H., Xu, X., Liu, C., Ren, T., Wu, K., Cao, X., Zhang, W., Yu, Y., Song, D. (2018). A machine learning approach to prevent malicious calls over telephony networks, *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 53–69.

Pandit, S., Liu, J., Perdisci, R., Ahamad, M. (2020). Fighting voice spam with a virtual assistant prototype.
`https://arxiv.org/abs/2008.03554`

Rodero, E. (2012). A comparative analysis of speech rate and perception in radio bulletins, *Text & Talk* **32**(3), 391–411.

Sahin, M., Francillon, A., Gupta, P., Ahamad, M. (2017). Sok: Fraud in telephony networks, *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 235–250.

Sawa, Y., Bhakta, R., Harris, I. G., Hadnagy, C. (2016). Detection of social engineering attacks through natural language processing of conversations, *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pp. 262–265.

Song, J., Kim, H., Gkelias, A. (2014). ivisher: Real-time detection of caller id spoofing, *ETRI Journal* **36**(5), 865–875.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A. et al. (2023). Gemini: a family of highly capable multimodal models, *arXiv preprint arXiv:2312.11805* .

Trapiņš, A. (2015). Krāpšanas gadījumu atklāšana un prevencija mobilo sakaru tīklos [Fraud detection and prevention in mobile networks].

WEB (a). European commission. Survey on "scams and fraud experienced by consumers".
`https://commission.europa.eu/system/files/2020-01/survey_on_scams_and_fraud_experienced_by_consumers_-_final_report.pdf`

WEB (b). Usa federal trade commission. Consumer sentinel network data book 2023.
`https://www.ftc.gov/system/files/ftc_gov/pdf/CSN-Annual-Data-Book-2023.pdf`

WEB (c). Finance latvia association. The amount of fraud prevented last year reached 9.2 million euros; the most difficult to prevent are cases of phone fraud.
`https://www.financelatvia.eu/news/pern-noversto-krapsanu-gadijumu-apmers-sasniedz-92-miljonus-eiro-visgrutak-noverst-telefonkrapsanas-gadijumus/`

WEB (d). Usa department of justice. Financial fraud crime victims.
`https://www.justice.gov/usao-wdwa/victim-witness/victim-info/financial-fraud`

WEB (e). Latvian state police. Fraudsters impersonate police officers and use emergency service phone numbers 110 and 112.
`https://www.vp.gov.lv/lv/jaunums/krapnieki-uzdodas-par-policistiem-un-izmanto-operativo-dienestu-talruna-numurus-110-un-112`

WEB (f). Google. Announcement of policy changes: April 6, 2022.
`https://support.google.com/googleplay/android-developer/answer/14554743`

WEB (g). Deepgram. The best speech-to-text apis in 2024.
`https://deepgram.com/learn/best-speech-to-text-apis`

WEB (h). Assemblyai. Industry's most accurate speech ai models.
`https://www.assemblyai.com/benchmarks`

WEB (i). Speechmatics. Introducing ursa from speechmatics.
https://www.speechmatics.com/company/articles-and-news/introducing-ursa-the-worlds-most-accurate-speech-to-text

WEB (j). Revai. Microsoft azure speech recognition vs. rev ai speech to text api.
https://www.rev.com/blog/resources/microsoft-azure-speech-recognition-vs-rev-ai-speech-to-text-api

WEB (k). Playht.
https://play.ht/

WEB (l). Anthropic. The claude 3 model family: Opus, sonnet, haiku.
https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf

Xing, J., Yu, M., Wang, S., Zhang, Y., Ding, Y. (2020). Automated fraudulent phone call recognition through deep learning, *Wireless Communications and Mobile Computing* **2020**(1), 8853468.

Zhao, Q., Chen, K., Li, T., Yang, Y., Wang, X. (2018). Detecting telecommunication fraud by understanding the contents of a call, *Cybersecurity* **1**, 1–12.

Received October 25, 2024 , revised March 3, 2025, accepted March 4, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, pp. 290–303
https://doi.org/10.22364/bjmc.2025.13.1.15

Comparison of Bayesian Neural Network Methods under Noisy Conditions

Rinalds Daniels PIKŠE, Evalds URTANS

Riga Technical University, Department of Artificial Intelligence and Systems Engineering, Riga, Latvia

rinalds.pikse@gmail.lv, evalds.urtans@rtu.lv

ORCID 0009-0005-4070-762X, ORCID 0000-0001-9813-0548

Abstract. This study presents an analysis of the effects of noise on the performance of Bayesian neural networks, comparing Monte Carlo Dropout (MC-D), Bayes by Backpropagation (BBB), and Variational Inference (VI) algorithms. The research aims to identify the strengths and weaknesses of each method in order to guide practitioners in selecting the most suitable method for various applications. The performance of the methods are evaluated using uncertainty estimates, accuracy, and robustness. This study gives insights into which method works best in noisy data and how well they can estimate the noise in the data. The results show that Bayes backpropagation is the most robust method in noisy conditions, while Monte Carlo Dropout is the most accurate in noise-free conditions. Variational Inference is the most sensitive to noise, but can be used as an estimator of the noisiness of the data set

Keywords: Machine Learning, Bayesian Neural Networks, Monte Carlo Dropout, Bayesian Inference, Bayes by Backprop, MC-D, VI, BBB

1 Introduction

Bayesian Neural Networks (BNNs) represent a significant advancement in neural network architectures, integrating Bayesian statistics to quantify predictive uncertainty. Unlike traditional networks, BNNs provide both point predictions and a probability distribution of outcomes, offering a more detailed assessment of model confidence. The ongoing development of BNN methodologies has substantially broadened their scope and utility, allowing more accurate estimation of weight uncertainties within the network (Jospin et al. (2020)).

This research specifically examines the most widely used methods of Variational Inference (VI), Monte Carlo Dropout (MC-D), and Bayes by Backpropagation (BBB).

These methods stand out for their substantial contributions and comprehensive theoretical underpinnings in uncertainty management. VI (Jordan et al. (1999)) enables the scalable approximation of complex posterior distributions. MC-D (Gal and Ghahramani (2016)) innovatively uses network dropout for Bayesian inference, offering an efficient alternative to traditional methods. Meanwhile, the BBB (Blundell et al. (2015)) seamlessly integrates Bayesian inference with the backpropagation process, providing a direct measure of weight uncertainty. The selection of VI, MC-D, and BBB reflects their pivotal roles in the evolution of BNN research and their adaptability to diverse application scenarios.

The aim of this study is to evaluate the performance of these three BNN methodologies in the presence of noise, a common challenge in real-world data environments. Noise, defined as random or irrelevant data that disrupt the learning process, can significantly impact the accuracy and reliability of the model. By introducing noise to the mushroom dataset, we simulate real-world data imperfections, enabling a comprehensive assessment of each BNN method's response to varying noise levels. This analysis seeks to identify the most effective BNN approach for managing uncertainty in noisy data, thereby informing practitioners' selection of appropriate methodologies for their specific applications. The purpose of this paper also is to find the methods that could be used to estimate the noise of the data set. The results of this study will provide valuable information on the relative strengths and limitations of VI, MC-D, and BBB, improving our understanding of the practical applicability of BNN in managing uncertainty in different data environments.

2 Related work

Recent developments in Bayesian neural networks (BNN) have introduced a variety of methods to integrate uncertainty quantification into deep learning. Jospin et al. (2020) provide a comprehensive classification of BNNs and a workflow for the design and implementation of BNN, focusing on the transition from conventional deep learning to Bayesian methods.

Specific research has also analyzed the performance of the methods used in this investigation. The Bayes backpropagation algorithm (BBB), introduced by Blundell et al. (2015), shows a performance comparable to the Monte Carlo dropout. This work underscores the effectiveness of BNNs in uncertainty estimation in different data scenarios. Olivier et al. (2021) investigate variational Bayesian inference (VI), highlighting the limitations of VI, and proposing enhancements for more accurate uncertainty quantification. Their work suggests the necessity of refining VI through alternative metrics and model averaging.

Other research has focused on investigating the relationship between uncertainty and "noisy" training data. Pawlowski et al. (2018) introduced "Bayes by Hypernet" (BbH), using hypernetworks as implicit distributions to model weight uncertainty. Their findings indicate that varying noise levels markedly affect predictive uncertainty, underscoring the importance of considering noise in BNN models. Furthermore, Shridhar et al. (2019) emphasizes the distinction between aleatoric and epistemic uncertainties for targeted model improvements, highlighting that aleatoric uncertainty is merely a

measure of noisy data. This distinction helps identify whether data quality or model limitations contribute to uncertainty, guiding strategies for improvement.

Building on these studies, our research compares Variational Inference (VI), Bayes by Backpropagation (BBB), and Monte Carlo Dropout (MC-D) against the mushroom dataset with controlled noise variations. By evaluating how each method responds to noise and quantifies uncertainty, this analysis aims to clarify their applicability and effectiveness in noisy conditions, contributing to the optimal selection of BNN methodologies for uncertainty management.

2.1 Variational Inference

In probabilistic models, inference refers to the process of estimating latent variables given observed data and the model's parameters. Latent variables, often referred to as hidden variables, are indirectly obtained from observable variables and influence the final result. The goal of variational inference is to create an approximate final posterior probability from observed and latent variables (Blei et al. (2016)).

Variational inference is a technique in Bayesian statistics in which an approximate distribution (the variational distribution) is used to represent the posterior distribution of the model parameters. The parameters of this variational distribution are optimized so that the distance between the variational distribution and the true posterior distribution is minimized, often using measures such as the Kullback-Leibler (KL) divergence, which measures the difference between two probability distributions based on Shannon information theory (Jospin et al. (2020)).

However, minimizing KL divergence is not easily applicable, as machine learning algorithms do not always have access to the true value distribution. Hence, using Jensen's inequality, it is possible to derive a new optimizable measure called ELBO (Evidence Lower BOund) (Blei et al. (2016)). Although minimizing the KL divergence is equivalent to maximizing ELBO, the advantage of the latter approach is that it does not require knowledge of the true value distribution for optimization (Jospin et al. (2020)).

Several methods can be used to maximize the ELBO value. One possible approach is the gradient descent method, which adjusts the weight parameters of the variational distribution to maximize the ELBO value. Stochastic Variational Inference (SVI) uses batches of samples that form probability distributions, which are iteratively improved using stochastic optimization (Hoffman et al. (2013a)). This approach can be scalable, since ELBO can be calculated for each batch of samples in each training iteration (Jospin et al. (2020)). ADVI is another method that seeks to maximize the ELBO value by transforming the inference problem into a uniform space and then solving the variational optimization problem. As described (Kucukelbir et al. (2015)), there are also black-box variational inference methods (Ranganath et al. (2014)), which allow maximizing ELBO, but often rely on stochastic optimization (Kucukelbir et al. (2015)), hence they can be considered a subgroup of SVI algorithms.

Structured Mean-Field Variational Inference (SMFVI) is a method that maintains dependencies between variables (Hoffman et al. (2013b)). This method is practical when variable relationships are known. Non-parametric models perform variational inference by creating a more complex value distribution that can contain several modes

(Nguyen and Bonilla (2013)), as illustrated in Figure 1. This figure provides an example of the multimodal problem, where a single-mode probability distribution would not fully characterize the true probability.

It is important to note that VI methods are not trained using backpropogation as other methods used in this research.

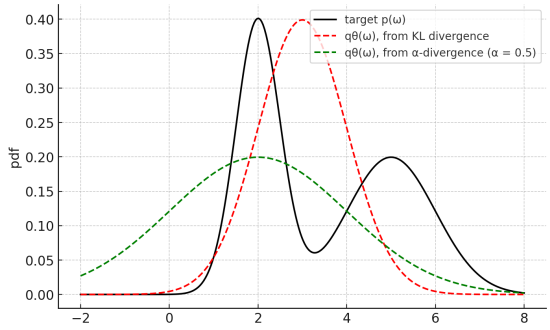


Fig. 1: Visualization of the multimodal problem, where a single-mode probability distribution would not fully characterize the true probability using KL divergence or alpha divergence.

2.2 Monte Carlo dropout

Monte Carlo Dropout (MC-D) is a method involving the stochastic deactivation of a proportion of neurons in a network. MC-D is widely adopted in artifical neural networks to avoid overfitting by reducing the influence of each individual neuron on the entire network. After model training, dropout is deactivated (Srivastava et al. (2014)).

This technique, where neurons are deactivated, can also be used in a different context, to calculate the uncertainty measure of a neural network. Gal and Ghahramani (Gal and Ghahramani (2016)) demonstrate that, by not deactivating dropout after training, this method can be considered an approximation of Bayesian methods and yield a distribution of values equivalent to the Variational Inference algorithm. Monte Carlo methods, through random neuron deactivation (dropout), allow the creation of a dynamic model that provides different results each time a specific data point is analyzed. This occurs because deactivated neurons are removed from the calculations, thus introducing variability in the model output after repeated evaluations.

MC-D is easily incorporated into existing artificial neural networks without the need for model retraining (Jospin et al. (2020)), making it a popular and practical choice for injecting Bayesian principles into established deep learning models. As a dropout-based approach, MC-D shares core functionalities with traditional artificial neural networks, including a layered neuron structure with adjustable weight and bias factors, the propagation of input data to deeper network layers, and iterative optimization of weight values via error backpropagation and optimization algorithms. Despite this, the uniqueness of MC-D within the Bayesian framework arises from its ability to reflect uncertainty via

multiple model evaluations, each influenced by a distinct dropout-induced network configuration.

MC-D consists of Dropout function that is used in training phase and testing phase. Normally, Dropout would be used only in training phase. In the training phase, dropout is used to prevent overfitting by randomly deactivating neurons. In the testing phase, dropout is used to approximate the posterior distribution of the model. This is done by running the model multiple times with different dropout configurations and aggregating the results. This method is used to estimate the uncertainty of the model. Dropout function is placed after the activation function in the neural network. The dropout rate is set to 0.5, which means that 50% of the neurons are deactivated.

2.3 Bayes by Backprop

Bayes by Backpropagation (BBB), an algorithm introduced in the research work "Weight Uncertainty in Neural Networks" (Blundell et al. (2015)), incorporates principles of variational inference in neural networks. Essentially, BBB performs an evaluation of neural network weights using distributions instead of fixed values. The goal of this approach is to introduce uncertainty into weight values, assessing the confidence in the model's prediction accuracy. Training is executed by maximizing the Variational Lower Bound (ELBO) cost function, using both the error function and Kullback-Leibler divergence for optimization. The authors of BBB argue that, based on their studies, this algorithm's classifying performance is comparable to the Monte Carlo dropout algorithm.

The BBB algorithm uses the reparametrization trick method to assess and optimize the weight distribution parameters in the neural network. Variable reparametrization has long been a technique used in the statistical literature, but only recently has it found applications in gradient-based machine learning (Kingma and Welling (2013)). This method proves useful when learning parameters that define the distribution of values, such as the distribution of weights in a neural network.

Within the Bayes by Backpropagation (BBB) algorithm, the weight distribution of the neural network is represented as a normal distribution with adjustable parameters μ and σ. During backpropagation, neural networks employ gradient descent, which requires the computation of gradients of the network error function relative to the weights of the network. These gradients guide the adjustment of the network's weights to minimize the error.

In the context of BBB, it becomes important to discern not only how the error fluctuates with respect to the weights of the network but also in relation to the parameters of the weight distribution. This presents a challenge, as computing gradients for stochastic values can be computationally complex and potentially lead to high variance due to inherent randomness. The reparameterization trick solves this issue - it reformulates the problem in a way that enables the calculation of gradients with respect to a deterministic weight distribution instead of a stochastic one, thus simplifying the gradient computation process.

This is achieved by initial sampling ϵ from a standard normal distribution, $N(0, I)$. This sample is subsequently transformed through the parameters of the weight distribution, μ (mean) and σ (standard deviation), to produce:

$$g_{\varphi}(\epsilon, x) = \mu + \sigma * \epsilon \tag{1}$$

In this equation, $g_{\varphi}(\epsilon, x)$ represents a deterministic function that models a sample from the weight distribution, facilitating the direct application of backpropagation for the computation of gradients. This deterministic transformation preserves the stochastic properties essential for Bayesian inference while streamlining the optimization process.

By adopting the reparametrization trick, BBB effectively transforms the stochastic variables into a format amenable to gradient-based optimization, combining the stochastic nature of Bayesian inference with the efficiency of deterministic gradient descent. Because of this, BBB can be comparable to a practical implementation of the Stochastic Variational Inference (SVI) algorithm with the reparametrization trick (Jospin et al. (2020)). A visual representation of the reparametrization trick method can be seen in Figure 2. To model probabilistic neural networks, the reparameterization trick is used to transform stochastic variables into deterministic ones. Deterministic values are necessary for backpropagation, which is used to optimize the weights of the neural network.

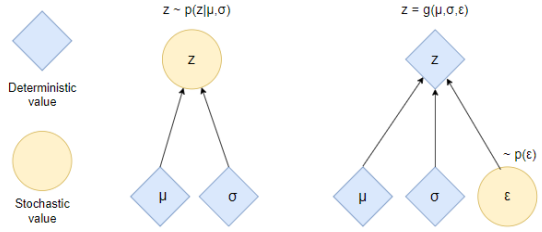


Fig. 2: Visualization of the reparametrization trick. In the middle of the image fully stochastic representation which is not derivable. In the right side of the image it is achieved by transforming the stochastic variable ϵ with the parameters μ and σ which are derivable.

3 Methodology

3.1 Formal definition of the task

In our experiments, we simulate real-world data noise by introducing it to both features and classes within our training data set. We use a custom program that randomly modifies a predetermined portion of the data. Specifically, for class noise, it flips the binary labels of a randomly selected subset of the data. For feature noise, it substitutes the original feature values with other legitimate values for a random set of records. This approach aims to mimic the variability and imperfections often found in real-world data, potentially impacting model predictions.

To evaluate our model's performance after adding noise, we focus on two metrics: accuracy and its standard deviation. Accuracy provides a quick measure of how often the model predicts correctly, often used in classification tasks. The standard deviation

of accuracy shows the model's consistency across tests, which is important for assessing stability in real-world scenarios. Together, they offer a concise evaluation of the effectiveness and reliability of a model.

We carried out six experiments, incrementally adjusting the noise rate from 0% to 50% in 10% steps.

3.2 Dataset

In this study, we employ the widely recognized mushroom classification dataset, sourced from the UCI Machine Learning Repository (Dua and Graff (2019)), consisting of 8,124 instances and 23 features that detail various species of mushrooms. This data set is particularly chosen for its balanced mix of 22 descriptive features and a binary classification column, poisonous (p) or edible (e), making it an ideal candidate for evaluating model performance in binary classification tasks. The inclusion of a diverse range of features, from 2 to 12 distinct categories each, introduces a level of complexity that mimics real-world data challenges, enhancing the relevance of our experiments. We converted categorical values to numerical codes to accommodate our data processing library's limitations, ensuring that the dataset's structure is preserved. The data set was randomly divided into training and testing subsets to ensure a representative sample. While using a single dataset might seem limiting, the mushroom dataset's balance between complexity and manageability uniquely positions it to explore the effects of noise injection on model performance. Other studies have not used this dataset, so our results will provide a fresh perspective on the performance of Bayesian neural networks in a binary classification task. In fact, other studies have not attempted to compare the performance of these three methods on the same dataset using methods that introduce noise to the data.

3.3 Applied approaches

The goal of our experiments is to compare the accuracy and standard deviation of the algorithms described with increasing levels of noise in the training data. During the experimental investigation, we have used the following implementations of the algorithms described in the following subsections.

3.3.1 Variational Inference A commonly used Python library for implementing Variational Inference is PyMC3 (Salvatier et al. (2015)), which provides a variety of tools for probabilistic programming and Bayesian analysis.

After preparing our dataset, a Bayesian model is first defined using the pm.Model() function of PyMC3. This function allows us to construct a Bayesian model that includes all prior required probability distributions and associated likelihood functions. Within this model, we assume normal distributions, frequently symbolized by alpha and beta, as our initial prior distributions.

The mean, μ, is estimated as a linear combination of the characteristics of the data set. This function considers the initial probability distribution values and the features

extracted from the dataset. Following this, the likelihood function is defined, typically employing the Bernoulli distribution, given the binary nature of many prediction tasks.

In Bayesian models employing VI, the prediction of mean is represented as a linear projection of the input features. For an input vector shown in Equation (2) the model typically defines the mean prediction as Equation (3) where w in Equation (4) is the weight vector (coefficients) associated with each feature and b is the bias term.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \tag{2}$$

$$\mu(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{d} w_i x_i + b \tag{3}$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_d]^T \tag{4}$$

In VI each weight w_i is treated as a random variable with an approximate posterior distribution. Generally, we assume a Gaussian variational posterior as shown in Equation (5) where mu_i is the mean (expected value) of the weight w_i, and σ_i^2 is the variance of the weight w_i.

$$q(w_i) = \mathcal{N}(w_i; \mu_i, \sigma_i^2) \tag{5}$$

Thus, when making predictions, one often uses the means of these weight distributions, resulting in the mean prediction being computed as shown in Equation (6).

$$\mu(\mathbf{x}) = \sum_{i=1}^{d} \mu_i \, x_i + b. \tag{6}$$

In this expression, the coefficients of the linear combination are the parameters μ_i, which are learned during the variational optimization process (typically by maximizing the Evidence Lower BOund, or ELBO).

The Variational Inference approximation is then performed using the pm.fit() function provided by the PyMC3 library. This step is crucial to the VI algorithm. It aims to find the optimal parameters that maximize the Evidence Lower BOund (ELBO), simultaneously working to minimize the Kullback-Leibler (KL) divergence.

Once the variational approximation is complete, we draw samples from the approximated posterior distribution. These samples allow us to generate predictions on an independent test data set. We evaluated the performance of the model by comparing these predictions with the actual outcomes of the test set. Metrics such as accuracy and the standard deviation of results provide insight into the performance and variability of the model.

3.3.2 Monte Carlo dropout In implementing the Monte Carlo dropout (MC-D) algorithm, we utilize PyTorch's torch.nn library (Paszke et al. (2019)). The execution of the algorithm occurs in two distinct phases: training and testing.

First, we define the model's linear layers using the torch.nn.Linear class. The data is then propagated through these network layers, resulting in a tensor for each record that contains two values representing each of the final classes.

The performance of the algorithm is then evaluated using a cross-entropy loss calculation. Additionally, we introduce dropout layers after the ReLU activation function applied to the first and second layers. By setting the dropout layers to a 70% rate, we mitigate overfitting and increase the model's ability to generalize across unseen data through the introduction of randomness.

During the testing phase, the Monte Carlo Dropout (MC-D) algorithm is executed 100 times on the same set of same input data for each test instance. This repeated execution strategy is designed to emulate the stochastic nature of dropout, allowing us to approximate the posterior distribution of model predictions. By aggregating the results of these 100 runs, we generate a distribution of predictions for each data point, from which we derive a measure of confidence or variance in the model results. For each run, we record the accuracy, specifically, the number of records correctly identified by the model, and compile these accuracy figures for graphical analysis. The aggregation of results after the predetermined 100 cycles provides a comprehensive overview of the model's performance consistency and its predictive confidence across the entire dataset.

The artificial neural network for the Monte Carlo Dropout (MC-D) algorithm comprises three layers: an input layer with 200 neurons, despite the dataset having 23 features, to allow for a richer feature representation through learned combinations; a hidden layer with 300 neurons to adequately learn data complexities without overfitting; and an output layer with 2 neurons corresponding to classification task. The neural network architecture has been chosen on the basis of preliminary testing to ensure optimal performance and learning efficiency. The number of layers is the same for all three methods to ensure a fair comparison.

We chose a learning rate of 0.001 and 30 epochs for training based on preliminary tests that showed an optimal balance between learning efficiency and avoiding overfitting. The 80%/20% training/test data split follows common practice for adequate learning and validation.

3.3.3 Bayes by backprop Bayes by backprop algorithm can be implemented using the PyTorch library, specifically using the torchbnn library (Lee et al. (2022)) for its Bayesian-specific functionalities. The neural network model is defined using the "Module" class from PyTorch, but the linear transformation layer is chosen from the "BayesLinear" function available in the torchbnn library. The activation function used is the ReLU function. An essential difference from other methods such as Monte Carlo Dropout (MC-D) is that, for each layer, the input and output data size variables are defined along with the values of μ and σ, which determine the distribution to define the weights of the model.

The BBB algorithm training phase includes calculating the KL divergence value in addition to the cross-entropy error. Both of these quantities are combined and used to

optimize the model to improve the total result. After the training phase, a testing phase is performed similarly to the MC-D algorithm, including error calculation, counting the correctly categorized records, and creating an error matrix.

Hyperparameter tuning is performed to determine the optimal configurations for the neural network. This adjustment establishes the number of neurons for each layer (typically 200 and 2 for the input and output layers, respectively), the μ and σ values for all layers (usually 0 and 0.1), and the weights for the KL divergence error and the cross-entropy error (0.3 and 0.7, respectively). The learning rate is set at 0.001 for 30 training epochs. The data set is divided into 80% for training and 20% for testing, with a batch size of 64, and 100 samples used for variance calculations.

4 Results

Results show that as the noise in the data set increases, the accuracy of all algorithms decreases, as seen in Figure 3. This observation is consistent with clean and noisy data for all three types of noise additions. These results are expected because, as the noise in the training data increases, the algorithm may start to emphasize random correlations instead of the true dependencies in the data. These findings highlight the importance of considering data noise when improving algorithm performance.

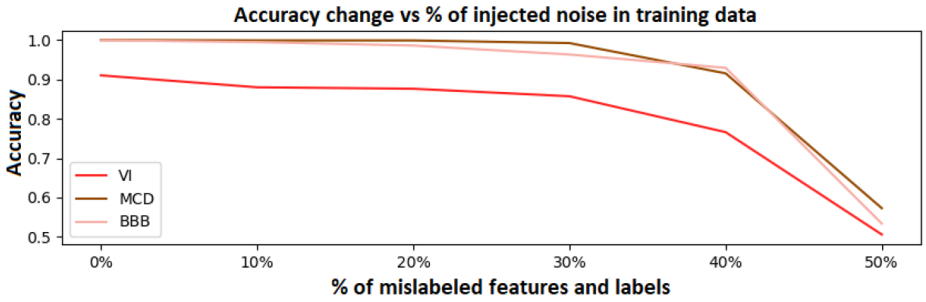


Fig. 3: Change in accuracy corresponding to the amount of added noise for each of the three algorithms.

Looking at the standard deviation values presented in Figure 4, there are noticeable differences between all algorithms. Variational inference (VI) exhibits the largest standard deviation, approximately 10 times greater than Bayes by Backprop (BBB) and noticeably larger than Monte Carlo Dropout (MC-D). This higher standard deviation for VI corresponds to its lower accuracy, suggesting that the algorithm's frequent errors tend to yield values far from the average. It is also noticeable that, for all algorithms, the standard deviation increases as the number of mixed labels grows, indicating that the algorithms are sensitive to the amount of noise. The largest changes in standard deviation with increasing noise are observed for the VI and MC-D algorithms.

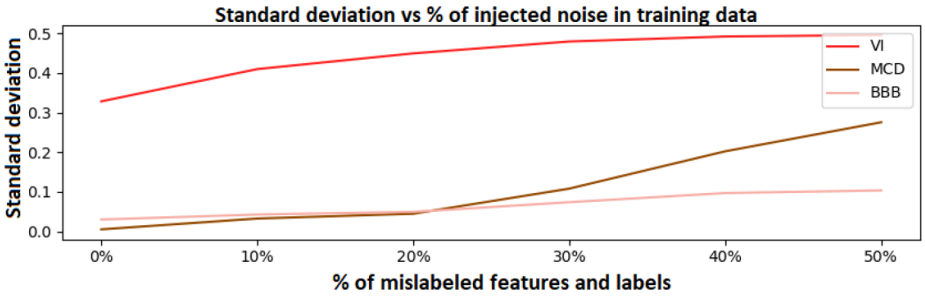


Fig. 4: Change in standard deviation in response to the quantity of incorporated noise for all three investigated algorithms.

Under conditions with 50% added noise, the highest accuracy and the lowest standard deviation are observed when noise is introduced exclusively to the features (Table 2). However, scenarios where noise is added solely to classes (Table 3) or to both classes and features (Table 1) yield similar accuracy results. This observation suggests differential impacts of class and feature noise on algorithm performance. It appears that output data noise may more strongly affect performance. A possible explanation is that, since the applied data set contains 23 different features, the algorithms are relatively more resistant to introducing noise into the features, using the information contained among the unmixed features. As seen in the tables below, in most experiments, MC-D showed the highest robustness with highest accuracy at different noise levels in both input features and output class labels. The negative correlation between precision and noise level is listed as $-r$, while the correlation between standard deviation and noise level is listed as r_{std}. From the results, it can be concluded that MC-D and BBB are more robust than VI, as these methods have a significantly lower correlation between accuracies and noise levels. On the other hand, VI and MC-D captures better noise in standard deviation as these methods have a higher correlation between noise.

Table 1: Accuracy and standard deviation of each method for different levels of noise in input features and class labels.

Method	0%	10%	20%	30%	40%	50%	$-r$	r_{std}
MC-D	**100** $\pm$ 0.00	**99.94** $\pm$ 0.03	**99.79** $\pm$ 0.05	**98.18** $\pm$ 0.10	90.67 $\pm$ 0.20	**59.49** $\pm$ 0.27	**0.77**	**0.97**
VI	90.76 $\pm$ 0.33	88.49 $\pm$ 0.41	87.62 $\pm$ 0.45	85.71 $\pm$ 0.48	76.60 $\pm$ 0.49	51.17 $\pm$ 0.50	0.84	0.93
BBB	99.70 $\pm$ 0.05	98.35 $\pm$ 0.06	98.05 $\pm$ 0.06	94.60 $\pm$ 0.07	**92.93** $\pm$ 0.11	55.29 $\pm$ 0.11	**0.77**	0.92

Table 2: Accuracy and standard deviation of each method for different levels of noise in input features only.

Method	0%	10%	20%	30%	40%	50%	$-r$	r_{std}
MC-D	**100** $\pm$ 0.01	**99.94** $\pm$ 0.02	**99.88** $\pm$ 0.02	**99.88** $\pm$ 0.03	**99.34** $\pm$ 0.05	**95.77** $\pm$ 0.07	**0.73**	0.95
VI	91.63 $\pm$ 0.33	88.49 $\pm$ 0.35	87.81 $\pm$ 0.36	86.88 $\pm$ 0.37	86.08 $\pm$ 0.37	84.36 $\pm$ 0.38	0.96	**0.96**
BBB	99.70 $\pm$ 0.05	99.04 $\pm$ 0.05	98.32 $\pm$ 0.05	95.68 $\pm$ 0.07	94.46 $\pm$ 0.06	92.33 $\pm$ 0.06	0.97	0.65

Table 3: Accuracy and standard deviation of each method for different levels of noise in class labels only.

Method	0%	10%	20%	30%	40%	50%	$-r$	r_{std}
MC-D	**100** $\pm$ 0.01	**100** $\pm$ 0.03	**99.94** $\pm$ 0.04	**98.71** $\pm$ 0.08	**94.72** $\pm$ 0.10	50.45 $\pm$ 0.09	0.72	**0.95**
VI	90.76 $\pm$ 0.33	91.87 $\pm$ 0.39	89.78 $\pm$ 0.44	90.95 $\pm$ 0.48	83.93 $\pm$ 0.49	53.39 $\pm$ 0.50	0.75	**0.95**
BBB	99.82 $\pm$ 0.05	99.70 $\pm$ 0.05	99.70 $\pm$ 0.06	98.38 $\pm$ 0.09	96.91 $\pm$ 0.12	**57.20** $\pm$ 0.09	**0.70**	0.84

5 Discussion

This study offers insights into the robustness of Bayesian algorithms, such as MC-Dropout (MC-D), Bayes by Backprop (BBB), and Variational Inference (VI), against data noise. Examination of their performance under varying degrees of data cleanliness underscores the necessity of assessing these algorithms across diverse datasets.

We also identify promising areas for further research. Enriching the comparison with additional Bayesian methods could refine our understanding, while studying Black Box Variational Inference or Structured Mean-Field Variational Inference methods might offer further insight into the relationships between VI, MC-D, and BBB algorithms.

Our findings highlight the significant impact of noise on accuracy and standard deviation. Future research should consider other potentially influencing factors, such as the complexity of the classification task and the size of the training dataset. Although BBB and MC-D showed comparable performance across test conditions, it would be worthwhile exploring circumstances under which one might outperform the other, thus informing algorithm selection in specific contexts.

The study has uncovered intriguing evidence that class mixing influences algorithm performance more than feature mixing. This observation could inform the strategic allocation of resources for data cleaning and help in balancing accuracy, dispersion, and execution time. We also suggest investigating the role of sample size in result dispersion and the resilience of different machine learning algorithms to data noise from various sources.

VI exhibits inferior performance under noisy conditions, primarily due to its estimation of the true posterior being extremely sensitive to data noise. The integration of hierarchical variational methods with annealed objectives may mitigate these issues, enhancing robustness against noise.

Lastly, an intriguing possibility emerging from our study is that algorithms could infer the level of noise in a dataset, presenting a potential avenue for cost savings, particularly in contexts where data cleaning expenses are significant.

6 Conclusions

This study sheds light on the sensitivity of Bayesian and Monte Carlo Dropout (MC-D) algorithms to data noise. Evidently, the MC-D and Variational Inference (VI) algorithms, in comparison to Backpropagation by Bayesian (BBB) methods, demonstrate an amplified response to noise, which could reflect a more precise uncertainty modeling.

In terms of binary classification tasks under various noise levels, both MC-D and BBB prove to be robust methods.

Performance differences become more pronounced under low-noise conditions, with both BBB and MC-D achieving high accuracy. In particular, MC-D maintains 100% accuracy even after deliberate noise introduction. In contrast, VI exhibits lower accuracy and, even with hyperparameter tuning, fails to match the performance of MC-D and BBB.

Significant differences also extend to dispersion values. With the increase of training data noise, VI and MC-D's dispersion is markedly affected, whereas BBB, despite showing changes, exhibits minor shifts, which raises questions about its ability to model confidence effectively. The dispersion that increases with noise allows for the estimation of noise in the dataset, which is a valuable feature in real-life datasets that are not always clean.

The study finds that up to 40% mixed training data, the algorithms under study can deliver high accuracy and stable dispersion. However, stability deteriorates and accuracy dips to approximately 50% when classes are mixed 50%. Interestingly, all three algorithms exhibit greater stability in feature mixing scenarios than in class label-only or combined class label-input feature mixing scenarios.

Highlighting the differential impact of noise, we observe that class-level noise in training data exerts a greater effect than feature-level noise. However, even with mixed 50% of all features, the algorithms maintain high accuracy and relatively low standard deviation. This stability is not mirrored when training data classes are mixed, emphasizing the importance of output noise over input noise in the training data.

References

Blei, D. M., Kucukelbir, A., McAuliffe, J. D. (2016). Variational inference: A review for statisticians, *Journal of the American Statistical Association* **112**, 859 – 877. `https://arxiv.org/abs/1601.00670`

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D. (2015). Weight uncertainty in neural networks, *ArXiv* **abs/1505.05424**. `http://proceedings.mlr.press/v37/blundell15.pdf`

Dua, D., Graff, C. (2019). Mushroom data set, `https://archive.ics.uci.edu/ml/datasets/mushroom`. UCI Machine Learning Repository.

Gal, Y., Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning., *in* Balcan, M.-F., Weinberger, K. Q. (eds), *ICML*, Vol. 48 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 1050–1059. `http://dblp.uni-trier.de/db/conf/icml/icml2016.html#GalG16`

Hoffman, M. D., Blei, D. M., Wang, C., Paisley, J. (2013a). Stochastic variational inference, *Journal of Machine Learning Research* **14**, 1303–1347. `http://www.jmlr.org/papers/v14/hoffman13a.html`

Hoffman, M. D., Blei, D. M., Wang, C., Paisley, J. (2013b). Stochastic variational inference, *Journal of Machine Learning Research* **14**, 1303–1347.
http://www.jmlr.org/papers/v14/hoffman13a.html

Jordan, M. I., Ghahramani, Z., Jaakkola, T., Saul, L. K. (1999). An introduction to variational methods for graphical models, *Machine Learning* **37**, 183–233.

Jospin, L. V., Buntine, W. L., Boussaid, F., Laga, H., Bennamoun (2020). Hands-on bayesian neural networks—a tutorial for deep learning users, *IEEE Computational Intelligence Magazine* **17**, 29–48.
http://arxiv.org/abs/2007.06823

Kingma, D. P., Welling, M. (2013). Auto-encoding variational bayes, *CoRR* **abs/1312.6114**.
http://arxiv.org/abs/1312.6114

Kucukelbir, A., Ranganath, R., Gelman, A., Blei, D. M. (2015). Automatic variational inference in stan, *NIPS*, pp. 568–576.
http://dblp.uni-trier.de/db/conf/nips/nips2015.html#KucukelbirRGB15

Lee, S., Kim, H., Lee, J. (2022). Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization, *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
https://ieeexplore.ieee.org/document/9761760

Nguyen, T. V., Bonilla, E. V. (2013). Efficient variational inference for gaussian process regression networks., *AISTATS*, Vol. 31 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 472–480.
http://dblp.uni-trier.de/db/conf/aistats/aistats2013.html#NguyenB13

Olivier, A., Shields, M. D., Graham-Brady, L. (2021). Bayesian neural networks for uncertainty quantification in data-driven materials modeling, *Computer Methods in Applied Mechanics and Engineering* **386**, 114079.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library, *NeurIPS*, Curran Associates, Inc., pp. 8024–8035.
http://dblp.uni-trier.de/db/conf/nips/nips2019.html#PaszkeGMLBCKLGA19

Pawlowski, N., Brock, A., Lee, M. C. H., Rajchl, M., Glocker, B. (2018). Implicit weight uncertainty in neural networks.
https://arxiv.org/abs/1711.01297

Ranganath, R., Gerrish, S., Blei, D. M. (2014). Black box variational inference.

Salvatier, J., Wiecki, T. V., Fonnesbeck, C. J. (2015). Probabilistic programming in python using pymc, *arXiv: Learning* .
https://arxiv.org/abs/1507.08050

Shridhar, K., Laumann, F., Liwicki, M. (2019). A comprehensive guide to bayesian convolutional neural network with variational inference, *ArXiv* **abs/1901.02731**.
http://arxiv.org/abs/1901.02731

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting., *Journal of Machine Learning Research* **15**, 1929–1958.
http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

Received September 7, 2023 , revised February 21, 2025, accepted March 3, 2025

Baltic J. Modern Computing, Vol. 13 (2025), No. 1, 304-314
https://doi.org/10.22364/bjmc.2025.13.1.16

Contemporary Higher EducationTeacher's Challenges. The Perspective on the AI in Studies

Vaiva ZUZEVICIUTE[1], Dileta JATAUTAITE[2], Edita BUTRIME[3]

[1] Mykolas Romeris University, Public Security Academy, Maironio str.7, Kaunas, Lithuania
[2] Vilnius University Business School, Sauletekio av. 22, Vilnius, Lithuania
[3] Lithuanian University of Health Sciences, A. Mickeviciaus str. 9, Kaunas, Lithuania

vaiva.zuzeviciute@mruni.eu, dileta.jatautaite@mv.vu.lt,
edita.butrime@lsmu.lt

ORCID 0000-0001-5768-1626, ORCID 0000-0003-4753-618X, ORCID 0000-0002-9795-4438

Abstract. Contemporary higher education teacher faces the challenges in their professional activities, which are both similar and a bit different than the challenges previous generations of teachers faced. This paper is dedicated to analysis of the situation and subjective interpretation of the situation by teachers in higher education. The paper aims at analyzing contemporary university teacher's challenges, including perspective on the AI in studies. The analysis is organized on the basis of three RQs. The methods of critical analysis of sources, analysis of personal experiences and a study based on qualitative methodology approach were employed for the development of this paper. The results of the theoretical analysis show that the challenges that contemporary HE teachers face are not unique, rather the degree of their manifestation is greater; in some cases the degree is more pronounced (e.g., IT (including AI) in studies) than in the other (e.g., marketization). The results of empiric study enabled conclusion that in some cases the theoretical considerations were empirically validated (teachers note challenge and opportunities prompted by technologies; pressure to publish), in some cases the challenges were not validated (marketization; multicultural setting as a challenge). Also empirical study enabled adding other challenges that a contemporary teacher faces on a regular basis: a challenge to deal with excessive, constant changes in legal stipulations and bureaucracy and a challenge to motivate students and help them grow.

Keywords: university/HE teacher; challenges; IT/AI in studies; marketization; multicultural setting.

Introduction

Contemporary higher education teacher faces the challenges in their professional activities, which are both similar and a bit different than the challenges previous generations of teachers faced. This paper is dedicated to analysis of the situation and subjective interpretation of it by teachers in higher education, because, while numerous studies and thus publications are dedicated to the situation, interpretations, feelings that

students face, not too many, however, are dedicated for at least the glimpse on how teachers are dealing in contemporary context (or at least studies again are linked to the feeling of students, James et al., 2019).

Terms *university* and *higher education* here are used as synonymous, though that may not apply for all countries; yet in significant portion of systems of education globally the synonymous usage is acceptable, if tertiary education in college or university is at focus. While *higher education* is a wider term, historically, *university* has more links, heritage and recognizability in the course of the last thousand years.

The paper **aims** at analyzing contemporary university teacher's challenges, including perspective on the AI in studies.

The analysis is organized on the basis of three **RQs**:

- What are the challenges (pressures) that a contemporary higher education teacher faces?
- What are the unique features of the contemporarily experienced challenges (as opposed to historically experienced)?
- What is the perspective of higher education teachers on those challenges (including the AI in studies)?

The **methods** of critical analysis of sources, analysis of personal experiences, and a study based on qualitative methodology approach were employed for the development of this paper.

HE Teacher's Challenges: Unique versus Universal

While we had thought that the historical horrors, such as the Holocaust, which wiped away into unimaginable suffering the entire Jewish communities, including devoted and enthusiastic university communities (Bacon, 2017), the contemporary events showed how mistaken we were to think about those horrors as something that we read about just in history textbooks. The authors of this paper discussed the theme, its importance for several months, more than a year into another human misery caused by a war in the Ukraine, which resulted in profound challenges for the Ukrainian higher education communities (Sytnykova et al., 2023; Ma et al., 2022). And, as if history teaches us nothing at all, the October 7th, 2023 brought back the worst memories, with the most horrific events happening again (https://www.foxnews.com/world/rocket-barrages-strike-southern-israel-operation-claimed-hamas), even if these are not directly related to higher education.

In the context of the suffering, profound challenges that either historically, or right here, right now university teachers had endured or are enduring, the theme of pressures that any other university teacher faces in a peaceful country seems mundane; and we feel a bit humbled and a bit ashamed to pursue it. Yet, human condition, while it is, unfortunately, still, even in 21st century, seems unavoidably linked to war, it is also linked to peace, to aspirations to live a happy, fulfilling life, to strive towards accomplishments, creativity, self realization, to aspirations to show one's best and to enjoy what best is out there in our civilization. Otherwise, human life will be void of any meaning, if the hope for progress is lost. Thus, even if in the face of recent events the challenges and crossroads for higher education teachers seem mundane, we will still

invite readers to join the conversation on what challenges we face today, and what is needed for us to be better equipped to face those challenges.

Contemporary higher education faces similar tasks that were there historically, just some of them are at a different degree; while in some cases the difference is minor, in other cases the magnitude of the degree is of different order thus making us think that the challenge is completely new. We are cautious about such classification.

The example of such difference is the fact that contemporary higher education is heavily influenced by globalization, which results in multi-cultural; multi-heritage classes. It may seem as a sign of contemporary developments, which undoubtedly add to pressures for a contemporary teacher (Zelenková and Hanesová, 2019). While we agree that the degree of multi-cultural representations is almost a given in almost any university, but historically, a university has been a multi-cultural hub for the last thousand of years (Zuzeviciute, 2011); just the degree is much more pronounced and visible. People from the far away territories (the term 'territories' is intentional: to denote the fact that several hundreds of years the states as we know them now did not exist or at least their borders shifted significantly) travelled to Bologna, then Oxford (starting at 11th, 12th c.), to Paris (starting at 13th c.), then to Krakow (starting at the 14th c.), because there were so few universities at that time. With Reformation and Contra-Reformation and also general civilizational advancements, the number of universities increased, thus the territorial needs for education started to be better, more readily accommodated in more accessible vicinities. Yet during the first several hundreds of years of the last millennium (thus at least 700 - 600 years ago) multi-cultural representation was a fact for those few operating universities that Europe had.

The next challenge, which seems associated just for the recent times is the fact that universities, due to their expansion and accessibility nationally and internationally in the last hundred years, became the market agents just as any other entity, competing for the market (Kind and Bunce, 2020; Praneviciene et al., 2017). These developments, due to profound changes triggered by the Reformation, Industrial revolution, Modernity are younger: spanning, just three - four hundred years. Reformation had a set of ideas, which are all of profound importance, but for our analysis the positive attitude and even encouragement for literacy (even if at first the literacy was valued as mainly a vehicle to read the Holy Book) is of most importance. The shift in thinking from literacy as something just for the few and probably not that good anyway, to thinking that literacy is important, changed the society immensely in the territories where Reformation ideas reigned. Then the ideas of Enlightenment joined, further strengthening the human potential, which resulted in advance of Modernity, with its science and technology, industrial revolutions (Zuzeviciute, 2011). While this paper does not hold ambitions to provide a comprehensive historical account, but rather it aims at illustrating the context of contemporary university. From being something rare, unique eight hundred years ago, in recent centuries it became absolutely necessary due to the elevated level of competencies necessary to run a highly sophisticated, technologized societies. The high number of universities, their relative accessibility, and mass university coverage however, led to the consequence of having to perform better than the next kid on the block. Hence: marketization, advertising, showing off one's colorful plumes. Thus, though this challenge is quite recent for the academia, teachers in higher education, but it became exasperated in recent decades. That happened due to the changing demographics, namely, the decreasing number of young people in societies, which have the longest history of modern universities, and the highest number on universities. Now,

the accomplishment of having many and accessible universities is becoming a liability, and, in order to proceed, the internationalization and global marketization are the two processes that joined forces to remediate the challenge of decreasing number of potential clients (that is: students). Thus higher education teachers today are more than ever encouraged, invited to participate in the activities that historically were not attributed to university teacher's activities. Such as participation in public events, in media and similar (Heck, 2022); while university teachers for hundreds of years had an active role in public life, but the contemporary change is not in the contents (still - to provide a valuable, evidence based, high quality consultation/opinion/expertise/insight), but in the purpose. That is, rather (or - at least next to) than provide valuable contents, it also has a purpose also to add to visibility of a given university. This task definitely adds to the pressure for a contemporary higher education teacher.

The third challenge that is constantly barraging a contemporary higher education teacher is the pressure to publish (Nguyen et al., 2021; MacPhail and O'Sullivan, 2019), which require some specific skills for academic writing (Hardy et al., 2022). But most importantly, in order to publish, a teacher has to participate in large, medium or at least small scale research in laboratory or in the field. No one argues the need to participate in research, the axiom itself was formulated by Humboldt two hundred years ago (here from Zuzeviciute, 2011). What is of concern, though: at times the balance, the efforts, the time, other resources are not being adequately distributed or the resources are scarce. Or the pressures outweigh resources to the point that an individual teacher feels abandoned and without a compass for the way forward with internal and unhelpful competition or/and administrative pressure starting dominating otherwise very important and rewarding aspect of university teacher's activities (Johansson, 2022). Though, most teachers find their way.

The fourth challenge is the changes and accessibility in available technologies. Again, seemingly, nothing new here, because historically technological advancements were intrinsically interwoven with other aspects of university life, because the very essence, the very purpose of university is to serve society via producing new knowledge and the ways to apply it, that is, technology. What changed, though, in recent decades is the relation to technologies. That is, the technologies, especially the IT based communication gained potential in the process of teaching and learning. That is, technologies, rather than being 'out there' in economy or –at best- supporting organisation of studies, gained the potential of actually becoming the medium or even an agent in teaching and learning. Yet, until the pandemic struck in 2020, many teachers avoided using IT too much, because of the respect for sensitive and facilitating human interaction. This was changed in a dramatic way during the almost two years of pandemic, with a short breathing break being again followed by introduction (invasion?) tools of AI entering the busy market in the end of 2022. Technological, methodological support higher education teachers (especially in social sciences, humanities, where the competence to use IT tools probably had to be expanded) received during pandemic (Bruggeman et al., 2022) was extremely important for their competence and confidence. Especially, when the targeted support was reinforced by peers' support (Gast et al., 2022). But, for many the achieved competence and confidence evaporated in the context of the new reality. Now, by the end of 2023 teachers are faced with questions, on the role of the AI tools in studies; on ethical considerations, on their own role in this new reality.

While we discussed at least four challenges (pressures?) that a contemporary higher education teacher faces with historical contextualization of those challenges, namely: 1) multicultural setting; 2) marketization;3) publication ('publish or perish'); 4) impact of IT, it seems warranted to reiterate that some of these challenges/pressures are not unique for contemporary teacher's experience.

Probably, with the exception of marketization (even in this case some historical examples may be provided on the opposite side of argument).

Multicultural setting, expectation to participate in research and publicize it, develop and offer for wider society's application of technologies always comprised the realities of a teacher at university. But – which deserves a special emphasis - the degree of manifestation of those challenges is noticeable.

Firstly, the depth and breadth of the challenges today are much more pronounced than they ever were.

Secondly, the number of teachers who are exposed to these challenges is almost universal, that is, for example: if previously (just a decade ago) we had some teachers who were very active in international study programmes, and some were less so, but today we will have to look twice for a teacher who does not participate in international study programmes in some way. Previously, just five years ago, we had some teachers who were using IT in their classes to a significant degree (e.g., Moodle for testing or for delivery of materials), but today we will have to look twice for a teacher who does not use the IT platforms in everyday work.

While marketization and publishing also used to comprise a reality of a teacher, today it is much more pronounced, yet the advent of AI tools is at the centre of contemporary discussions among academia.

In order to validate at least to some degree theoretical considerations and the analysis of our personal experiences, the study was designed and implemented in 2023.

Methodology and procedure

The international study was implemented during 2nd-3rd quarters of 2023. The study is based on qualitative research methodology; the set of questions was developed on the most pressing issues that colleagues (teachers in higher education) may face. The purpose of the study was to validate whether the four challenges (multicultural setting; expectation to publish; expectation to participate in marketization; impact of IT/AI in studies) are among the challenges that teachers note as impactful challenges. The interviews were conducted by the authors either face to face or using el. formats; the anonymity of participants was safeguarded by eliminating the identifiers and by shuffling the responses in the phase of data analysis (Content analysis was used).

Participants

Totally 16 participants shared their ideas, impressions on what is exciting, difficult, interesting for them in their work. 2 representatives from Romania, 2 from Portugal, 2 from Poland, 1 from Latvia, 4 from the Ukraine, 1 from Lithuania, 1 from the Check Republic, 1 from Italy, 1 from Finland, 1 from India shared their perspective.

The colleagues were well established in their careers: experience of higher education teacher spanned from more than 50 years to 9 in that capacity (more than 50;

37; 34; 30; 29; 27 (2); 28; 25; 18; 17 (2); 15; 13; 11; 9). Teachers teach: logics; cultural industries; economics; management; social work; psychology; foreign languages; social sciences; informatics, natural sciences, climate studies, human resources development.

Thus quite a comprehensive coverage in terms of the experience in higher education, in terms of the subjects taught and in geographical coverage (from 10countries), thus even if the qualitative approach will always result to certain degree of **limitation**, but the wide geographical and thematic coverage serves as a compensatory factor.

In total, colleagues were invited to share their ideas on 9 questions. If participant provided more than one idea, the ideas were counted, rather than participants, thus the N of answers/categories is not set: the N for some questions may exceed the total number of participants. Due to the limitations for the scope of the paper only a part of data is shared here (the answers to 5 questions out of 9 questions).

Results

The first result that was observable from the very start was the stark difference between the answers from colleagues in all the 9 countries and then the Ukraine, thus resulting in creating in some cases almost two sets of categories. Namely, the colleagues, experiencing war for almost two years repeatedly indicated '*war'*, '*working under air raids*' as the challenges they have to overcome every day and every hour. The challenge was not taken into consideration during theoretical analysis, which only reinforces two of the axioms of research: firstly, we are all biased, even as researchers who invest specific and focused effort to maintain objectivity. E.g., in this case, while representing a country, which is not in war, we did not think of this challenge. Secondly, while war rages, many of the things, which are taken for granted are being re-evaluated and judged differently.

However, the Ukrainian colleagues, while indicated stress of war, always used 'but', and still provided answers to the questions according to the logics of questions, thus their answers are not excluded, but included into the analysis.

To the question *What is the most exciting thing for you at this moment in your profession (as a teacher in Higher education)?* 4 main categories of answers were identified (Table 1).

Table 1. What is the most exciting thing for you at this moment in your profession (as a teacher in Higher education)?

Categories	**Entries**
New technologies	6 (2 of them specifically indicated AI) entries
Complementing teaching and research, especially, while involving students	3 entries
Working with creative, motivated, innovative, enthusiastic and empathetic students	3 entries
Developing/creating new course-units/curriculum, which is reflective of the contemporary developments	2 entries

One participant elaborated on the optimism as based on the experiences of the pandemic, when the IT was used so productively (thus making optimistic expectations for AI in studies logical). Also *partnership and student's autonomy and creativity* was noted by one of the participants as one of the most exciting issues in higher education.

Regarding the next question: *What is the most difficult thing for you at this moment in your profession (as a teacher in Higher education)?* 3 main categories were identified (Table 2).

Table 2. What is the most difficult thing for you at this moment in your profession (as a teacher in Higher education)?

Categories	Entries
New IT technologies in class	8 (2 of them specifically: to organize remote work under the duress of air-raid); 1 of them specifically on using AI) entries
Motivate students	6 (3 of them: specifically under the constant stresses posed by war; 1 of them: specifically on-line studies during pandemic, especially for girls) entries
Additional bureaucracy (reporting; additional paper work without substantiating its purpose or need, internal 'politicking')	5 entries

Two more participants' indicated *new application of educational methods* without specifying whether the application is related to IT, thus the category is outlined as a separate one. One colleague mentioned the problem of *low attendance*, which leads to difficulties for motivating students and their intellectual growth, also, one colleague, indicated *no problems/difficulties*. Another contributor identified *developing and offering students a motivating, creative tack that is conducive to learning* as a demanding task.

While sharing their ideas on question *What other influences/factors/events/processes do you identify today that have an impact on your work as a teacher in higher education today?* participants identified several factors, which were categorized as follows (Table 3).

Table 3. What other influences/factors/events/processes do you identify today that have an impact on your work as a teacher in higher education today?

Categories	Entries
Technologies in class	8 entries
Constant changing in legislative requirements and additional work outside the class	4 entries

Colleagues also noted *constant stress caused by war* (2 entries) and s*tress to keep up with smart/educated/worldly students* (2 entries).

The question *How do you feel today as a teacher in Higher education after the pandemic, the AI influence and other influences/factors/events/ processes/changes?* was less objectivity orientated as it supplied a possible focus for the answer.

Interestingly, the answers supplied at times completely opposite opinions (the first **three** categories) (Table 4).

Table 4. How do you feel today as a teacher in Higher education after the pandemic, the AI influence and other influences/factors/events/ processes/changes?

Categories	Entries
This is exciting, and I am looking forward to what it will bring to studies	4 entries
That is and will be a huge challenge	3 entries
I am not yet aware of the AI having a role in studies	2 entries
This is so new I am stressed out and I lack support	2 entries
I feel exhausted and overwhelmed and think on giving up the profession	1 entry

Two more entries equaled challenges to *war* experiences without specific focus on the AI, which illustrates the notion discussed earlier in the paper: those extreme duress changes perspectives and the priorities. One more contribution emphasized the *need to help student to get ready for different and - possibly – disadvantaged eventualities*, which may be related to the contributions on war.

While teachers, our colleagues in 10 countries were asked to share ideas on *What other influences do you identify that make your work more challenging/difficult, that make a positive impact on how you perceive your profession?* several expected, but also unexpected responses were received, which were categorized as follows (Table 5).

Table 5. What other influences do you identify that make your work more challenging/difficult, that make a positive impact on how you perceive your profession?

Categories	Entries
New technologies in general have a positive impact	5 entries (3 of them specifically on AI)
New generation of technology savvy colleagues	3 entries
Changes in all fields, including professional, thus making lifelong learning an absolute necessity (thus a burden on balancing work, family life)	2 entries
Excessive bureaucracy	2 entries

One colleague shared the similar concerns, regarding *unhelpful policies, excessive bureaucracy*, but added that these factors do not hinder the enjoyment of teaching. Also, the pride in one's university's resilience and revival even under the duress of war was indicated; the issue of *remuneration* and as well the *excessive and changing legal requirements* were mentioned. Another contributor indicated that even seemingly negative factors, such as *excessive bureaucracy, new technologies may give (and are giving) opportunities* for personal and professional growth.

The empiric study enabled to collect subjective perspective of teachers/colleagues in several countries on theoretically identified considerations. In some cases enriching perspectives were received, in some cases the theoretically formulated challenges did not receive any validation from participants, in some cases the validation was received (Figure 1).

Challenges that were validated:	
Teach and encourage learning of students within the context of AI	The challenge was definitely validated in empirical study
Need to participate in research and publication	The challenge was validated to some small degree empirically
Challenges that were not validated:	
Expectation to participate in marketization	The challenge was not at all validated empirically
Need to work in a multicultural setting	The challenge was not at all validated empirically
The empirical study enabled identification additional challenges that contemporary higher education teachers face	
Excessive and constantly changing legal regulations and additional tasks (bureaucracy)	
Motivate students (taking into account low attendance, necessity to develop creating engaging tasks)	

Figure 1. The validated, non-validated and added challenges for a contemporary higher education teacher

The challenge related to overcoming war were indicated several times, but it was not indicated as a separate challenge, because it is obviously self-explanatory.

Conclusions

Theoretical analysis (critical analysis of sources and analysis of personal experiences of authors) enables formulation of some conclusions.

Within historical contextualization, a contemporary higher education teacher faces at least four major challenges/pressures: 1) the need to work in a multicultural setting; 2) the expectation to participate in marketization; 3) the need to participate in research and publication; 4) teach and encourage learning of students within the context of AI. Some of these challenges/pressures are not unique for contemporary teacher's experience. While a multicultural setting, and expectation to participate in research and publicize it

always was among intrinsic expectations for a teacher at university, however, marketization may be indicated as a more recent one.

However, the degree of manifestation of those challenges/pressures is different. Firstly, the depth and breadth of the challenges today are much more pronounced than they ever were. Secondly, the number of teachers who are exposed to these challenges and have to perform in the face of those challenges is almost universal as opposed to just some proponents and enthusiasts.

The results of empirical study enable formulating conclusions on the perspectives of teachers in HE.

In some cases, the theoretical considerations were empirically validated:

Technologies (entries were indicated among the answers to almost all of the interview questions; e.g., for question What other influences/factors/events/processes do you identify today that have an impact on your work as a teacher in higher education today: 8 entries were allocated in the category indicating the fact that technologies became the integral part in classes).

Pressure to publish was mentioned (e.g., answers to the question: What is the most exciting thing for you at this moment in your profession (as a teacher in Higher education)? – 3 entries mentioned excitement to complement teaching and research, especially, involving students).

In some cases they were not validated (marketization; multicultural setting as challenges were not mentioned at all).

The fact that marketization was not identified may hypothetically be explained by the assumption that administrative apparatus at universities take upon this task, and that indeed the publishing of research findings comprises marketing function to major degree in the academia. The findings that teachers did not indicate multicultural setting as a challenge were unexpected; the authors of the paper do not have explanation for this finding.

Empirical study enabled adding other challenges that a contemporary teacher faces on regular bases:

The challenge to deal with excessive, constant changes in legal stipulations and excessive bureaucracy (e.g., answers to the question: What other influences/factors/events/processes do you identify today that have an impact on your work as a teacher in higher education today? 4 entries indicated frustration with constant changes in legislative requirements and additional work outside the class).

The challenge to constantly motivate students and help them grow (e.g., answers to the question: What is the most exciting thing for you at this moment in your profession (as a teacher in Higher education)?: 3 entries reflected the need to help students to maintain enthusiasm with their studies, to create challenging, motivating tasks, etc.).

Acknowledgements

We appreciate the input of Dr. Nataliia Yemets at the Chernihiv Polytechnic National University, Department of Psychology and Creative Industries for assisting in collecting data from the Ukraine.

References

Bacon, E. K. (2017). *Saving Lives in Auschwitz: The Prisoners Hospital in Buna-Monowitz*. Purdue University Press. DOI: 10.2307/j.ctv15wxppb OCLC: 1000143211 LCCallNum: D806 .B33 2017

Bruggeman, B., Garone, A., Struyven, K., Pynoo, B., Tondeur, J. (2022). Exploring university teachers' online education during COVID-19: Tensions between enthusiasm and stress. *Computers and Education Open*, 3, 100095.

Gast, I., Neelen, M., Delnoij, L., Menten, M., Mihai, A., Grohnert, T. (2022). Supporting the well-being of new university teachers through teacher professional development. *Frontiers in Psychology*, 13, 866000. ISSN: 1664-1078 EISSN: 1664-1078 DOI: 10.3389/fpsyg.2022.866000 PMID: 35967696

Hardy, A., Murray, R., Thow, M., Smith, M. (2022). 'So maybe I'm not such an imposter': becoming an academic after a life as a teacher-practitioner. *Higher Education Research & Development*, 41(1), 163-176. DOI: 10.1080/07294360.2020.1835835

Heck, D. (2022). Teacher educators as public intellectuals: exploring possibilities. Asia-Pacific *Journal of Teacher Education*, 50(2), 118-129. DOI: 10.1080/1359866X.2022.2049700

James, C., Strevens, C., Field, R., Wilson, C. (2019). Student wellbeing through teacher wellbeing: A study with law teachers in the UK and Australia. *Student Success*, 10(3), 76-83. https://doi.org/10.5204/ssj.v10i3.1338

Johansson, T. (2022). Do evaluative pressures and group identification cultivate competitive orientations and cynical attitudes among academics? *Journal of Business Ethics, 176*(4), 761-780. DOI: 10.1007/s10551-020-04670-7.

King, N., Bunce, L. (2020). Academics' perceptions of students' motivation for learning and their own motivation for teaching in a marketized higher education context. *British Journal of Educational Psychology*, 90(3), 790-808. DOI: 10.1111/bjep.12332 PMID: 31814108

Ma, X., Gryshova, I., Koshkalda, I., Suska, A., Gryshova, R., Riasnianska, A., Tupchii, O. (2022). Necessity of Post-War Renewal of University Teachers' Potential in Terms of Sustainable Development in Ukraine. *Sustainability*, 14(19), 12598. ISSN: 2071-1050 EISSN: 2071-1050 DOI: 10.3390/su141912598

MacPhail, A., O'Sullivan, M. (2020). Challenges for Irish teacher educators in being active users and producers of research. In *Teacher Educators as Teachers and as Researchers* (pp. 64-78). Routledge. DOI: 10.1080/02619768.2019.1641486

Nguyen, N., Pham, L., Cox, S., Bui, N. (2021). Departmental Leadership and Peer Pressure on Academic Research Performance at Universities in Emerging Countries: An Empirical Study in Vietnam. *Journal of University Teaching and Learning Practice, 18*(6), 119-134. DOI: 10.53761/1.18.6.09

Praneviciene, B., Zuzeviciute, V., Vasiliauskiene V., Simanaviciene Z. (2017). Internationalization of higher education: Lithuanian experience in Bologna Process and Beyond. *Montenegrin journal of economics. Podgorica : Economic Laboratory Transition Research Podgorica*-Elit, 2017, vol. 13, no. 1.

Sytnykova, Y., Shlenova, M., Kyrpenko, Y., Kyrpenko, V., Konoplenko, N., Hrynchenko, I. (2023). Teaching Technologies Online: Changes of Experience in Wartime in Ukraine. *International Journal of Emerging Technologies in Learning*, 18(18). DOI:10.3991/ijet.v18i18.40491

Zelenková, A., Hanesová, D. (2019). Intercultural competence of university teachers: a challenge of internationalization. *Journal of Language and Cultural Education*, 7(1), 1-18. DOI: 10.2478/jolace-2019-0001

Zuzeviciute, V. (2011). *Learning at university: challenges, strategies, perspectives for lifelong learning*. Saarbrücken: Lambert academic publishing.

Received June 7, 2024, revised March 15, 2025, accepted March 16, 2025