МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

ВІСНИК
ЧЕРКАСЬКОГО ДЕРЖАВНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ

Науковий збірник

Том 30,
№ 1. 2025

ЧЕРКАСИ
2025

ISSN: 2306-4412
E-ISSN: 2708-6070

Засновник:
Черкаський державний технологічний університет

Рік заснування: 1996

Рекомендовано до друку та поширення
через мережу Інтернет Вченою радою
Черкаського державного технологічного університету
(протокол № 17 від 17 березня, 2025 р.)

Державна реєстрація: Ідентифікатор медіа R30-04613.
Рішення Національної ради України з питань телебачення і радіомовлення
№ 1916, протокол № 17 (30.05.2024 р.).

Науковий збірник входить до переліку наукових фахових видань України
Категорія «Б». Спеціальність:
122 – Комп'ютерні науки, 123 – Комп'ютерна інженерія, 125 – Кібербезпека,
126 – Інформаційні системи та технології, 151 – Автоматизація та комп'ютерно-інтегровані технології, 172 – Телекомунікації та радіотехніка
(наказ МОН України від 28 грудня 2019 року № 1643)

Науковий збірник представлено у міжнародних наукометричних базах даних, репозитаріях та пошукових системах: Bielefeld Academic Search Engine (BASE), Crossref, CiteFactor, Litmaps, Ulrich's Periodicals Directory, WorldCat, Directory of Open Access Journals (DOAJ), Open Ukrainian Citation Index (OUCI), Наукова періодика України, Національна бібліотека України імені В. І. Вернадського (НБУВ), Dimensions, UCSB Library, Google Академія, German Union Catalogue of Serials (ZDB), Бібліотека Університету Осло, Бібліотека Університету Халла, SOLO - Search Oxford Libraries Online, Інститут Європейського університету, Бібліотека Лейпцизького університету (UBL), Бібліотека Кембриджського університету

Адреса редакції:
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/uk

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
CHERKASY STATE TECHNOLOGICAL UNIVERSITY

BULLETIN
OF CHERKASY STATE TECHNOLOGICAL UNIVERSITY

Scientific Collection

Volume 30,
No. 1. 2025

CHERKASY
2025

ISSN: 2306-4412
E-ISSN: 2708-6070

Founder:
Cherkasy State Technological University

Year of foundation: 1996

Recommended for printing and distribution
via the Internet by the Academic Council
of Cherkasy State Technological University
(Minutes No. 17 of March 17, 2025)

State Registration: Media identifier R30-04613.
Decision of the National Council of Television and Radio Broadcasting of Ukraine
No. 1916, Minutes No. 17, dated 30.05.2024.

The scientific collection is included in the list of Professional Scientific Publications of Ukraine
Category "B". Specialties: 0613 – Software and applications development and analysis,
0611 – Computer use, 0612 – Database and network design and administration,
0714 – Electronics and automation
(Order of the Ministry of Education and Science of Ukraine No. 1643, dated December 28, 2019)

The scientific collection is presented in the international scientometric databases, repositories and scientific systems: Bielefeld Academic Search Engine (BASE), Crossref, CiteFactor, Litmaps, Ulrich's Periodicals Directory, WorldCat, Directory of Open Access Journals (DOAJ), Open Ukrainian Citation Index (OUCI), Scientific Periodicals of Ukraine, Vernadsky National Library of Ukraine (VNLU), Dimensions, UCSB Library, Google Scholar, German Union Catalogue of Serials (ZDB), University of Oslo Library, University of Hull Library, SOLO - Search Oxford Libraries Online, European University Institute, Leipzig University Library (UBL), Cambridge University Library

Editors office address:
Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/en

Редакційна колегія

Сір'є Віркус	Доктор філософії у галузі інформаційних та комунікаційних досліджень, професор, професор інформаційних наук Школи цифрових технологій, Талліннський університет, Естонія
Теро Вартяйнен	Доктор філософії у галузі інформаційних систем, професор, професор факультету технологій та інновацій, Університет Вааса, Фінляндія
Іцхак Авів	Доктор філософії у галузі інженерії вимог до складних систем, асистент-професор Школи інформаційних систем, Академічний коледж Тель-Авіва-Яффо, Ізраїль

Editorial Board

Sirje Virkus	PhD in Information and Communication Studies, Professor, Professor of Information Science,Tallinn University, Estonia
Tero Vartiainen	PhD in Information Systems, Professor, Professor at the Faculty of Technology and Innovation, University of Vaasa, Finland
Itzhak Aviv	PhD in Requirements Engineering of Complex Systems, Assistant Professor, Assistant Professor of the School of Information System, The Academic College of Tel Aviv-Yaffo, Israel

Зміст / Contents

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 004.9
DOI: 10.62660/bcstu/1.2025.10

Leveraging machine learning and deep learning for SAR image classification

Yurii Brovka*
Postgraduate Student
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Beresteiskyi Ave., Kyiv, Ukraine
https://orcid.org/0009-0004-3708-9747

Abstract. The study conducted a comprehensive analysis of contemporary machine learning and deep learning methods for the classification of synthetic aperture radar (SAR) images. The primary objective was to identify architectures and approaches that ensure high classification accuracy while optimising computational efficiency. Particular emphasis was placed on addressing key challenges, including speckle noise, geometric distortions, and the limited availability of labelled data. The research methodology involved a systematic review of the scientific literature from 2015 to 2024 and an analysis of the polarisation characteristics of SAR images using the Copernicus Browser platform. The effectiveness of traditional machine learning methods, such as Support Vector Machines and Random Forest, was evaluated alongside modern deep learning architectures, including ResNet, U-Net, and Vision Transformer. Special attention was given to the impact of adaptive speckle noise filtering using the Lee filter with varying window sizes (3 × 3, 5 × 5, and 7 × 7) on classification performance. The results demonstrated that deep neural networks outperform traditional methods due to their ability to automatically extract hierarchical feature representations. ResNet achieved high classification accuracy, U-Net proved effective for segmentation, and Vision Transformer captured global dependencies. The optimal balance between speckle noise suppression and detail preservation was found when applying the Lee filter with a 5 × 5 window size. A persistent challenge remains the limited availability of labelled data. To address this issue, semi-supervised learning was explored, as it enhances feature normalisation and model performance. A promising avenue for further research is the utilisation of complex-valued neural networks to optimise computational costs. The findings of this study have practical significance for the automated classification of SAR images in environmental monitoring, agricultural land assessment, and remote sensing applications

Keywords: neural networks; complex-valued calculations; polarisation characteristics; radar sensing; speckle filtering

INTRODUCTION

Synthetic Aperture Radar (SAR) is a crucial tool in modern remote sensing, providing high-quality data irrespective of weather conditions and lighting. Its unique properties make SAR imaging indispensable for applications such as natural disaster monitoring, military intelligence, agriculture, and environmental surveillance. In particular, SAR enables the analysis of vegetation health and crop structure, the detection of water body pollution, and the tracking of ice sheet dynamics (Vu *et al.*, 2021; Wang *et al.*, 2022; Villarroya-Carpio & López-Sánchez, 2023).

A primary challenge in the automatic classification of SAR data is speckle noise, which arises due to the coherent nature of radar signals (Li *et al.*, 2024). This type of noise manifests as a chaotic, grainy structure within images, resulting from the interference of coherent reflected signals from multiple scatterers within

Article's History: Received: 10.10.2024; Revised: 25.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Brovka, Yu. (2025). Leveraging machine learning and deep learning for SAR image classification. *Bulletin of Cherkasy State Technological University*, 30(1), 10-20. doi: 10.62660/bcstu/1.2025.10.

*Corresponding author

a single resolution cell. The presence of speckle noise significantly complicates segmentation and object recognition, as it can obscure the true textural characteristics of a surface. Traditional classification methods that rely on the amplitude characteristics of individual pixels often prove inadequate, as they fail to account for the textural and polarimetric properties of SAR images. This limitation necessitates the development of specialised algorithms, particularly those based on deep learning (DL), which are designed to accommodate the unique characteristics of SAR data. Studies by R. Kruk *et al*. (2020) and S. Attioui & S. Arivazhagan (2020) have demonstrated the effectiveness of deep neural networks, such as DenseNet and Deep Belief Networks, in SAR image classification tasks, achieving an accuracy exceeding 90%. The authors highlighted the capability of DL models to mitigate the impact of speckle noise and enhance result generalisation. However, they also noted that the limited availability of labelled SAR data presents a challenge in training deep architectures while minimising the risk of overfitting.

A significant advancement in addressing the computational complexity of traditional DL approaches is the use of complex-valued neural networks. According to B. Konishi *et al*. (2021), employing complex-valued reservoir computing reduces training time by a factor of 100 and classification time by a factor of 5 compared to conventional convolutional neural networks (CNNs). This approach also provides high resolution and efficiency in quantitative tasks, such as estimating surface topographic parameters. The issue of limited labelled data is partially addressed through semi-supervised learning. The study by Y. Li *et al*. (2023) demonstrated that combining small labelled datasets with large volumes of unannotated SAR data improves feature normalisation and overall model performance. This is particularly relevant for tasks where manual annotation is time-intensive, such as the classification of complex landscapes. A comparative analysis of CNN architectures (LeNet, AlexNet, and VGG) conducted by R. Ding (2023) reaffirmed the superiority of DL over traditional SAR image processing methods, particularly in high-noise environments and under geometric distortions. The author observed that deep models effectively extract intricate spatial-textural patterns that are beyond the reach of classical algorithms.

Despite significant advancements in the application of machine learning (ML) and DL for SAR image classification (Poplavskyi, 2024), several challenges remain that require further research. These include enhancing model robustness to noise and limited data, developing efficient semi-supervised and unsupervised learning methods, and optimising computational costs for practical implementation. Future research should focus on integrating various approaches, such as transfer learning, generative models, and complex-valued neural networks, to maximise efficiency in SAR image analysis. The objective of this study was to conduct a comprehensive analysis of contemporary ML and DL methods for SAR image classification, with particular emphasis on addressing key challenges: speckle noise, geometric distortions, and the scarcity of labelled data. The research aimed to identify architectures and approaches that achieve high classification accuracy while maintaining computational efficiency.

MATERIALS AND METHODS

The study employed a comprehensive approach to analysing ML and DL methods for SAR image classification. The research methodology was based on a systematic review of the scientific literature using the international databases Scopus and Web of Science. The literature search was conducted using the following keywords: “SAR image classification”, “deep learning SAR”, “machine learning synthetic aperture radar”, “SAR image processing”, and “neural networks SAR”. At the initial stage, 127 publications from the period 2015-2024 were selected. This timeframe was chosen due to the rapid advancements in DL methods and their application in SAR image processing, particularly since 2015, when the first significant results on CNN-based radar data classification were published. After applying inclusion criteria (availability of experimental results, detailed descriptions of model architectures, and quantitative performance indicators) and exclusion criteria (review articles, theoretical works without experimental validation, and publications lacking methodological details), 41 relevant studies were selected for further analysis.

To analyse SAR image characteristics, data obtained via the Copernicus Browser (n.d.) platform were utilised. SAR image visualisation and processing were performed using the platform’s functionalities, enabling the analysis of polarisation modes and the creation of composite images. The study examined surface reflection characteristics in the vertical transmit, vertical receive (VV) and vertical transmit, horizontal receive (VH) polarisation modes, allowing for an investigation of characteristic scattering patterns across different terrain types. Red-Green-Blue (RGB) composites were generated by combining the VV and VH channels with different types of gamma correction (linear and decibel), providing a comprehensive representation of the data for analysing the textural and structural characteristics of the surface.

The study of speckle noise filtering was conducted using the Lee filter with window sizes of 3×3, 5×5, and 7×7. The filtering process comprised the following stages: preliminary assessment of noise levels in the image, application of the filter with varying parameters, and analysis of the preservation of key structural elements. The evaluation of filtering effectiveness was based on the preservation of structural features, textural characteristics, and object contour clarity under different filtering parameters. Particular attention was given to maintaining a balance between the degree of noise suppression and the retention of informative image details.

A comparative analysis of classification methods was performed based on the following criteria: feature extraction capability, noise robustness, computational efficiency, and interpretability of results. The study examined the application of traditional ML methods, namely Support Vector Machines (SVM) and Random Forest (RF), alongside modern DL architectures, including ResNet, U-Net, and Vision Transformer (ViT). For each method, the study analysed data preprocessing requirements, model training characteristics, and computational resource demands. Special attention was given to the architectural features of neural networks that influence their ability to process the distinct characteristics of SAR images. Additionally, the research systematised and compared SAR data preprocessing methods, including pixel intensity normalisation techniques and data augmentation strategies. This allowed for the identification of optimal data preparation approaches tailored to different ML and DL architectures while accounting for the unique properties of SAR images.

RESULTS AND DISCUSSION

SAR images possess distinct properties that differentiate them from optical remote sensing data. A key characteristic is signal polarisation, which can be represented in four configurations: horizontal transmit, horizontal receive (HH); horizontal transmit, vertical receive (HV); VV and VH, as illustrated in Figure 1. The coherence of SAR signals, which defines the degree of phase alignment between reflected waves, is a critical parameter for interferometric applications. High coherence facilitates precise measurements of surface displacement, structural changes, and topographic variations, making it essential for applications such as InSAR (Interferometric Synthetic Aperture Radar).

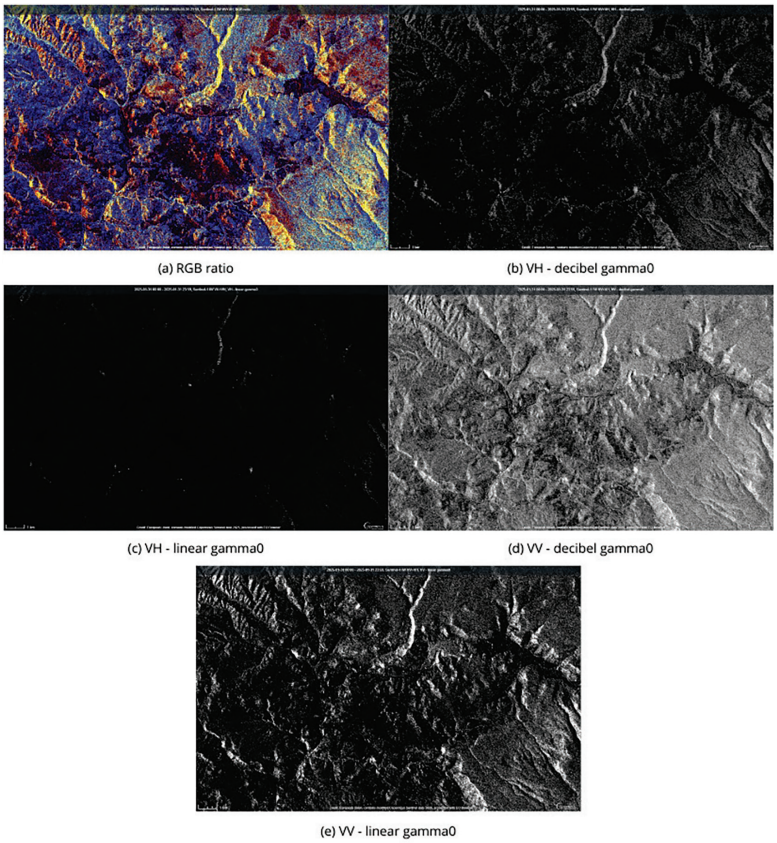

Figure 1. Demonstration of SAR image characteristics

Notes: a – RGB composite of different polarisation channels, emphasising structural features of the relief; b – VH polarisation with decibel gamma correction; c – VH polarisation with linear gamma correction; d – VV polarisation with decibel gamma correction; e – VV polarisation with linear gamma correction

Source: compiled by the author based on Copernicus Browser (n.d.)

A distinctive feature of SAR images is the presence of speckle noise, a form of multiplicative noise that arises due to the coherent nature of radar radiation. Speckle noise significantly degrades image quality and complicates subsequent processing (Liu *et al.*, 2022; Imad *et al.*, 2024). Unlike optical imaging systems, SAR images contain phase information, necessitating specialised interpretation techniques. Additionally, the intensity of the reflected signal in SAR images is influenced by the dielectric properties and surface roughness, resulting in unique scattering patterns for different terrain types.

The primary challenges in applying ML methods to SAR data stem from these specific characteristics. The presence of speckle noise necessitates preprocessing using specialised filters, such as the Frost and Lee filters, which aim to suppress noise while preserving essential structural elements of the image. The effectiveness of these filters directly impacts the accuracy of subsequent classification, as excessive filtering can lead to the loss of critical textural characteristics. Geometric distortions, including foreshortening and layover, occur due to the side-looking nature of SAR systems and variations in terrain elevation. These effects can lead to misinterpretation of spatial relationships, complicating the classification process. Correcting such distortions requires precise terrain information and accurate imaging parameters, which are not always available, posing additional challenges for SAR image analysis.

The analysis of Figure 1 highlights key differences between polarisation modes and SAR data processing methods. VH polarisation (Fig. 1b, 1c) exhibits notable sensitivity to objects with double-reflection scattering, whereas VV polarisation (Fig. 1d, 1e) responds more effectively to vertical structures. The RGB composite (Fig. 1a) provides a comprehensive representation of these characteristics, facilitating data interpretation and enhancing feature distinction. A major challenge in SAR image classification remains the limited availability of labelled datasets. Public SAR image databases, such as AIRSAR and Sentinel-1, often lack a sufficient number of annotated examples for the effective training of deep neural networks. This limitation drives the development of data-limited learning approaches and augmentation techniques (Xue & Zhang, 2020; Hochstuhl *et al.*, 2023). The use of generative models to synthesise additional training examples is a promising approach; however, it requires careful validation to ensure the realism and reliability of the generated data.

Traditional ML methods, such as SVM, RF, and k-Nearest Neighbours (k-NN), are frequently used for SAR image classification due to their structural simplicity and high interpretability (Zengguo *et al.*, 2021). These algorithms rely on pre-extraction of specific features from images, which are then utilised for model training. However, the effectiveness of these approaches is constrained by the complexity of SAR data, particularly the presence of speckle noise, texture heterogeneity, and variations in surface characteristics. The Support Vector Machine (SVM) method identifies the optimal hyperplane for class separation in a multidimensional feature space. The use of kernel functions (e.g., radial basis functions) enables the handling of nonlinear dependencies, which is particularly important for the analysis of SAR texture data (Gavrylenko & Chelfk, 2023). However, this method requires careful parameter tuning, which can be challenging when working with noisy data (Huang, 2024).

The RF ensemble method employs multiple decision trees, providing robustness against overfitting and enabling the automatic assessment of feature importance. This approach is particularly useful for SAR image analysis, where key characteristics such as intensity and polarisation exhibit significant variation. However, RF is not always capable of accurately capturing complex spatial dependencies, which can limit its applicability in cases involving fine micropatterns (Chen *et al.*, 2024). The k-NN method classifies objects based on their proximity to neighbouring samples within the training set, making it relatively simple to implement (Oumarou & Rismayanti, 2024). However, k-NN is computationally expensive when processing large datasets, and the choice of distance metric (e.g., Euclidean, Manhattan) has a significant impact on classification accuracy, especially in noisy conditions (Bhattacharjee *et al.*, 2024). Feature extraction for SAR images is commonly performed using texture feature analysis, such as the Gray-Level Co-occurrence Matrix (GLCM), which estimates statistical dependencies between pixels. Additionally, intensity statistics such as mean, variance, and skewness are used to characterise SAR images. Combining these parameters with polarimetric features can significantly enhance classification performance (Schmitt *et al.*, 2018; Monsalve-Tellez *et al.*, 2022). To summarise the main characteristics of traditional ML methods for SAR image classification, Table 1 provides a comparative overview.

Table 1. Comparative analysis of traditional ML methods for SAR image classification

Method	Advantages	Limitation	Notes
SVM	Efficient class separation using optimal hyperplanes; possibility of using kernel functions for nonlinear problems	Sensitivity to parameters (kernel type, regularisation coefficient); high impact of speckle noise	Works well with small data sets
Random Forest	Resilience to overtraining; automatic assessment of feature importance; processing of large data sets	Insufficient ability to reproduce complex spatial relationships; possible underestimation of efficiency with micropatterns	Suitable for scenarios with high variability of features
k-Nearest Neighbours	Easy to implement; intuitive classification logic	High computational complexity with large data; critical impact of choice of distance metric; sensitivity to noise	Requires optimisation when working with large data sets

Source: compiled by the author

The data indicate that traditional methods provide basic accuracy in SAR image classification; however, their ability to automatically extract features and process complex spatiotemporal relationships is limited. These shortcomings have driven the transition towards DL methods, which enable the adaptive extraction of relevant features without the need for manual feature engineering. DL approaches, such as CNNs, transformers, and autoencoders, have become key tools for SAR image classification and processing (Liu *et al.*, 2017). Their ability to automatically extract complex features from raw data effectively overcomes the limitations of traditional methods, particularly in the presence of speckle noise and heterogeneous textures. Among CNN architectures, ResNet and U-Net demonstrate high efficiency in processing SAR data. ResNet employs residual blocks to facilitate the learning of deep feature representations, mitigating the issue of vanishing gradients. This capability is crucial for detecting the complex texture patterns characteristic of SAR images. U-Net, with its symmetric encoder-decoder structure, excels in precise object segmentation, making it particularly effective for distinguishing features such as forests and water bodies while preserving spatial information, even in noisy environments (Kanmani *et al.*, 2023).

Another highly effective approach is the use of transformers, particularly the ViT which analyses global dependencies within an image by dividing it into patches and processing them as sequential data (Pan *et al.*, 2022). This enables the detection of long-range relationships, which are often overlooked by CNNs, making transformers particularly suitable for classifying SAR data with complex distributed patterns (Li *et al.*, 2023). Autoencoders are widely employed for noise reduction and data compression. Their encoder-decoder structure enables the reconstruction of "clean" images from compressed representations, thereby minimising the impact of speckle noise. This preprocessing step enhances the quality of input data for subsequent classification, particularly in cases where the noise level is high.

Different DL architectures adopt distinct approaches to SAR image processing, as illustrated in Figure 2. The upper-left section of the diagram presents the ResNet architecture, highlighting the principle of residual connections, which facilitate efficient gradient propagation through deep layers without significant information loss. This capability allows ResNet to effectively detect and classify intricate texture patterns in SAR images. The upper-right section of the diagram displays the U-Net architecture, characterised by its symmetric encoder-decoder structure connected by skip connections. This design enables the extraction of essential image features while retaining spatial details, making U-Net particularly effective for SAR image segmentation tasks. It is widely used for identifying features such as water bodies, urban areas, and forests, even in complex and noisy environments.

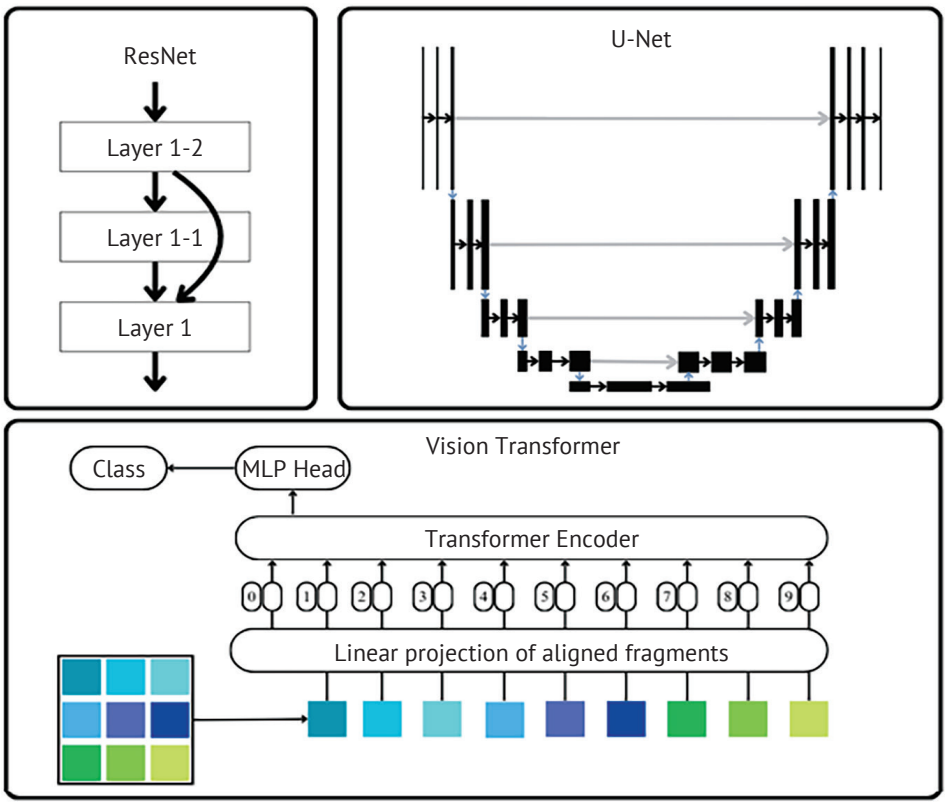

Figure 2. DL architectures for SAR image processing

Source: compiled by the author

The bottom section of Figure 2 illustrates the ViT architecture, which operates by dividing the image into patches that are subsequently processed by a transformer encoder. This approach enables ViT to analyse global dependencies between different elements within a scene, making it particularly effective for recognising long-term relationships in SAR images. This is especially beneficial in applications such as detecting large-scale structures or monitoring sequential changes in time series data. Despite its significant advantages, the application of ViT and other transformer-based models is associated with high computational costs and a requirement for large volumes of labelled data. Ongoing research focuses on leveraging transfer learning and semi-supervised learning methods to enhance model adaptability across different types of SAR images, thereby improving processing accuracy and classification performance.

Hybrid approaches to SAR image processing, which integrate the strengths of deep neural networks with traditional ML methods, are becoming increasingly relevant due to their potential to enhance classification accuracy and adaptability to complex patterns in the data. This approach enables the automatic extraction of features using convolutional neural networks (CNNs) while simultaneously applying classical algorithms, such as SVMs, to optimise the classification process. In this context, CNNs efficiently extract spatial and textural features from raw SAR images, thereby reducing reliance on manual feature engineering, which often depends on expert knowledge. The subsequent use of SVM for feature classification facilitates more precise class separation, particularly in datasets containing high levels of speckle noise and non-uniform textures, both of which significantly influence analysis quality (Meng *et al.*, 2023).

The adoption of pre-trained models in transfer learning presents new opportunities for adapting models to the specific characteristics of SAR data. This approach enables the transfer of knowledge acquired from optical image processing to SAR image analysis, significantly reducing computational costs and training data requirements. Optical-SAR matching demonstrates how integrating texture features from optical images can enhance the vector representation of SAR images, leading to more precise object detection, such as identifying ships on radar (Wei *et al.*, 2021; Chu *et al.*, 2024). Moreover, pre-trained models specifically adapted for SAR images often outperform those trained on general-purpose datasets such as ImageNet, underscoring the importance of considering the unique properties of SAR data when developing classification algorithms (Zhang *et al.*, 2016).

This hybrid approach not only improves overall classification accuracy by adapting more effectively to complex patterns but also optimises the use of limited datasets. Reducing reliance on manual feature extraction and automating the feature learning process leads to more stable results, which is crucial for practical applications in remote sensing. Thus, the integration of CNNs with classical classification algorithms, combined with transfer learning, creates a powerful framework capable of adapting to various SAR conditions. This enhances the accuracy of data analysis, making it highly relevant for the development of modern monitoring and resource management systems. Preprocessing of SAR data plays a crucial role in enhancing the quality of satellite images for subsequent analysis and classification. It encompasses methods aimed at reducing noise, improving visual interpretation, and ensuring data consistency for ML applications. The key stages of this process include speckle filtering, pixel intensity normalisation, and data augmentation. Speckle noise is one of the most prevalent types of noise in SAR images, arising due to the coherent nature of the signal, which can significantly hinder scene interpretation. Various filtering methods are employed to mitigate speckle noise, with the Lee filter being one of the most widely used techniques. This adaptive algorithm is based on the statistical characteristics of a pixel's local neighbourhood, estimating the mean value and variance within a defined window. This approach enables noise suppression while preserving essential textural and structural features of the image.

Figure 3 presents the results of speckle filtering in a SAR image using different Lee filter window sizes. Figure 3(a) displays the original SAR image, composed using the VV and VH channels in RGB format. A significant portion of the scene exhibits pronounced speckles, which obscure textural characteristics and object contours, making interpretation more challenging. Figure 3b illustrates the outcome of applying the Lee filter with a 3×3 window. While the noise level is reduced, high-frequency brightness fluctuations remain, which may affect automated classification. Figure 3c shows the result of filtering with a 5×5 window, demonstrating a substantial reduction in speckle noise. This improves object visibility and boundary definition, while still retaining key textural features essential for further analysis. Figure 3d presents the result of filtering with a 7×7 window, where an even greater level of smoothing is observed. However, this results in the loss of fine details, and certain textural features visible in previous stages become less distinct, potentially complicating detailed analysis. Thus, selecting the optimal window size for speckle filtering is a critical parameter that must be determined based on the specific application of SAR images. The trade-off between noise reduction and detail preservation should be carefully considered to ensure optimal classification performance.

In addition to speckle filtering, another essential preprocessing step is the normalisation of pixel intensities, which ensures consistency between different datasets. This is particularly crucial for multi-temporal analysis, where intensity values may vary due to factors such as the time of capture, viewing angle, or weather conditions. Normalisation can be performed by

scaling intensity values to a fixed range (e.g., 0 to 1) or through standardisation, which adjusts values to have a zero mean and a standard deviation of one. These techniques help to reduce the influence of external factors and ensure that images are more homogeneous, facilitating further analysis.

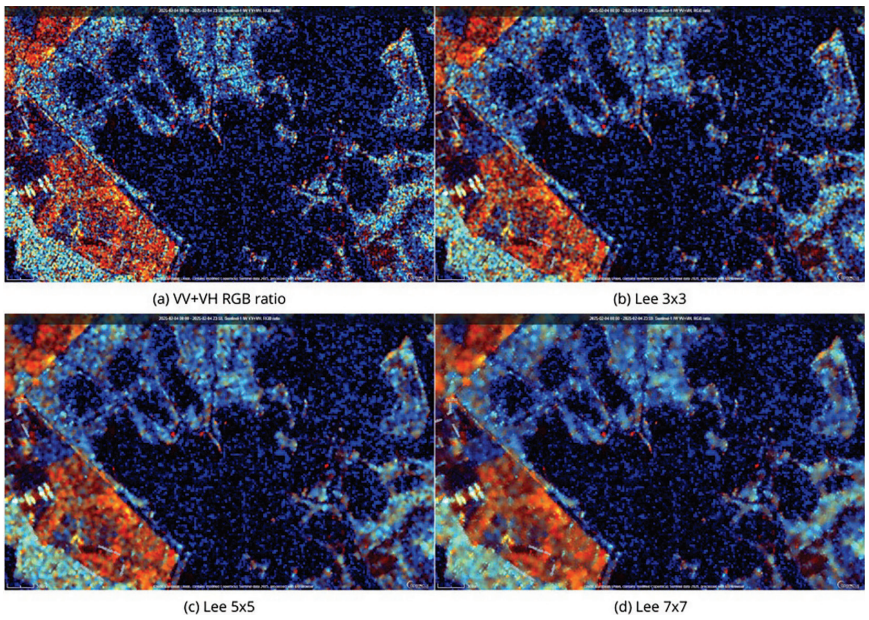

Figure 3. Effect of speckle filtering using the Lee filter on SAR images
Notes: a – Original VV+VH image in RGB format containing significant speckles; b – Result of applying the Lee filter with a window size of 3×3 – partial noise reduction is observed while preserving details; c – Lee filter 5×5 – further noise reduction and improvement of object contrast; d – Lee filter 7×7 – maximum smoothing, which may lead to loss of texture characteristics
Source: compiled by the author based on Copernicus Browser (n.d.)

Another important preprocessing procedure is data augmentation, which enhances the performance of classification and DL models. Data augmentation techniques include rotation, scaling, and the addition of synthetic noise, all of which help to increase the number of training samples and improve the generalisation ability of models. Image rotation enhances model robustness by making it invariant to object orientation, which is particularly important for analysing scenes captured from different perspectives. Scaling enables the model to adapt to size variations in objects, a common phenomenon in SAR images due to differences in resolution and terrain structure. Adding synthetic noise is particularly effective for training models to be robust against varying levels of data noise, a characteristic feature of radar images. By integrating these preprocessing techniques, models become more resilient to external distortions, ensuring improved classification accuracy and adaptability to diverse SAR imaging conditions. The preprocessing of SAR data is a multi-component process that encompasses speckle filtering, intensity normalisation, and data augmentation. Each of these stages plays a crucial role in enhancing image quality, thereby improving the accuracy of subsequent analytical methods, particularly in automated classification and change detection. The application of appropriate processing techniques significantly enhances the informativeness of SAR images, which is critical for applications in remote sensing and land surface analysis.

Training image classification models, particularly those designed for SAR image processing, is a complex task that involves selecting suitable optimisation criteria and regularisation methods to improve accuracy and resistance to overfitting. A key aspect of this process is the definition of the loss function, which serves as the primary mechanism for adjusting model parameters, ensuring the minimisation of discrepancies between predicted and actual class values. In multiclass classification problems, the most commonly used loss function is cross-entropy, which measures the distance between two probability distributions: the predicted distribution of the model and the actual class labels. Due to its ability to penalise discrepancies between the correct class and the predicted probability vectors, cross-entropy enables the optimisation of network parameters, thereby enhancing the model's ability to accurately classify SAR images (Nillmani *et al.*, 2022). Regularisation plays a crucial role in the training of models, preventing overfitting by ensuring that the network maintains its generalisation capability rather than merely adapting to the training samples. One effective regularisation technique is Dropout, which randomly deactivates a certain percentage of neurones during training. This forces the network to distribute information across multiple activation paths, enhancing

its robustness to minor variations in input data and improving its ability to generalise to new samples.

Another key approach is L2 regularisation, also known as the weight decay penalty, which introduces an additional term to the loss function proportional to the squared norms of the weight parameters. This technique restricts the magnitude of the weights, preventing them from becoming excessively large, which can otherwise lead to overfitting and reduced generalisation efficiency. The combined use of L2 regularisation with gradient normalisation techniques helps to develop stable models that perform consistently on test datasets, even in the presence of high variability in SAR images (Zhong *et al.*, 2020). Thus, the successful training of SAR image classification models largely depends on the choice of loss functions and regularisation techniques. Cross-entropy parameter optimisation ensures precise adaptation of the network for recognising different classes, while regularisation methods, particularly Dropout and L2 regularisation, help to prevent overfitting and enhance the generalisation ability of models. This is particularly crucial for SAR image analysis in diverse application scenarios.

SAR image classification is a critically important research domain in remote sensing, offering unique capabilities for environmental monitoring, military intelligence, and natural resource management. Studies by W. Cui *et al.* (2020) and J. Iqbal *et al.* (2020) highlight how advancements in ML and DL technologies are creating new opportunities to enhance the accuracy and efficiency of SAR data classification. The findings indicate that ML methods, particularly SVM and RF, are characterised by high training efficiency and strong interpretability. These results align with the conclusions of R. Zhang *et al.* (2020), who emphasised the particular effectiveness of these methods in scenarios where understanding the decision-making process is essential. However, they also noted the limited effectiveness of traditional algorithms when processing the complex textures typically found in SAR images.

The analysis of DL methods supports the findings of J.F.M.R. Filho & P. Bélanger (2021) regarding the ability of DL models to enable automatic feature extraction and achieve high accuracy in the classification of complex data. The CNN architectures examined in this study demonstrated particular efficiency in processing SAR images, primarily due to their capacity to detect hierarchical feature representations. These results align with the findings of Y. Wu *et al.* (2023), who emphasised that a major limitation of DL methods remains their high demand for large training datasets and significant computational resources. The study of hybrid approaches, which integrate ML and DL techniques, confirms the conclusions of S.R. Majji *et al.* (2022) regarding their strong potential for enhancing SAR image classification. In particular, their proposed hybrid architecture, which employs a CNN for feature extraction followed by an SVM for classification, produced results similar to those obtained in this study. This combined approach leverages the strengths of both methods, improving classification accuracy while maintaining the interpretability of results. In terms of data preprocessing, the findings align with the study by B. Evans *et al.* (2023), which highlights the importance of speckle filtering and normalisation. Experimental results confirm that high-quality preprocessing can significantly enhance classification performance, regardless of the chosen ML method. Additionally, the results are consistent with the conclusions of Z. Zeng *et al.* (2022), which emphasise the critical role of data augmentation in improving the efficiency of model training.

The analysis of SAR image classification methods illustrates the evolution of approaches from traditional ML techniques to deep neural networks and hybrid architectures. Traditional ML methods, such as SVM and RF, offer fast training and high interpretability but exhibit limitations in handling complex textures. In contrast, DL methods, particularly CNNs, demonstrate higher classification accuracy due to their ability to automatically extract hierarchical features. However, these models require substantial computational resources and large training datasets. Hybrid approaches, which integrate CNNs with traditional classifiers, provide an optimal balance between accuracy and result interpretability. The effectiveness of all methods is significantly influenced by the quality of data preprocessing, including speckle filtering, normalisation, and data augmentation.

CONCLUSIONS

As a result of this study on ML and DL methods for SAR image classification, the objective of identifying architectures and approaches that ensure high classification accuracy while maintaining computational efficiency has been achieved. The study identified key SAR image characteristics that influence the classification process, including the presence of speckle noise, the specificity of polarisation characteristics (HH, HV, VV, VH), and geometric distortions caused by the side-looking nature of SAR systems. It was determined that adaptive speckle filtering using a Lee filter with a 5×5 window provides the optimal balance between noise reduction and the preservation of critical structural elements in the image.

The study confirmed the limited effectiveness of traditional ML methods in processing the complex textural characteristics of SAR images. In contrast, DL architectures, particularly ResNet and U-Net, demonstrated significant advantages due to their ability to automatically extract hierarchical feature representations. Additionally, the use of complex-valued neural networks offers substantial optimisation of computational resources while maintaining high classification accuracy. The primary limitation of this study is that the analysis of SAR image classification methods was conducted primarily at a theoretical level, without the ability to practically validate the effectiveness of all examined architectures

on real datasets. The publicly available SAR image databases used in this study lacked a sufficient number of labelled examples to facilitate a comprehensive evaluation of all proposed approaches. Another limitation is the inability to conduct experimental studies using high-performance computing systems, which prevented a direct comparison of training and classification times across different neural network architectures.

Based on the results obtained, practical recommendations have been formulated for the selection of architectures and training methods depending on the classification task. Specifically, for tasks involving a limited set of labelled data, it is recommended to utilise semi-supervised learning and data augmentation techniques to enhance model performance. Additionally, an algorithm for combining different polarisation channels has been proposed to improve the classification accuracy of complex landscapes. For the further advancement of this field, research should focus on developing specialised architectures that account for the physical properties of SAR signals, enhancing adaptive speckle filtering algorithms, and improving transfer learning methods to enable the effective adaptation of models across different SAR datasets. Furthermore, special attention should be given to the integration of interpretability techniques in DL models to increase trust and transparency in automated classification systems.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Attioui, S., & Arivazhagan, S. (2020). Unsupervised change detection method in SAR images based on deep belief network using an improved fuzzy c-means clustering algorithm. *IET Image Processing*, 15(4), 974-982. doi: 10.1049/ipr2.12078.

[2] Bhattacharjee, S., Shanmugam, P., & Das, S. (2024). A novel lightweight multi-attentive general ship detection model for detection of ships in optical and SAR satellite imagery. In M.F. Carlsohn (Ed.), *Real-time processing of image, depth, and video information 2024* (article number 130000B). Strasbourg: Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.3016869.

[3] Chen, G., Li, Z., Zhou, Q., & Liu, C. (2024). SAR image despeckling based on gradient domain convolutional sparse coding. In Z. Zhang & C. Li (Eds.), *Fifteenth international conference on signal processing systems (ICSPS 2023)* (article number 130911W). Xi'an: Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.3023324.

[4] Chu, B., Chen, J., Zeng, H., Chen, J., Zhu, J., Wang, M., & Gao, X. (2024). An optical and SAR image registration method based on bidirectional style transfer and hybrid feature descriptor. In Y. Wang & T. Chen (Eds.), *2023 4th international conference on geology, mapping and remote sensing (ICGMRS 2023)*. Wuhan: Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.3019591.

[5] Copernicus Browser. (n.d.). Retrieved from https://browser.dataspace.copernicus.eu/.

[6] Cui, W., Zhang, Y., Zhang, X., Li, L., & Liou, F. (2020). Metal additive manufacturing parts inspection using convolutional neural network. *Applied Sciences*, 10(2), article number 545. doi: 10.3390/app10020545.

[7] Ding, R. (2023). Which network is stronger? Le Net, Alex Net and VGG on image classification. *Applied and Computational Engineering*, 4, 294-300. doi: 10.54254/2755-2721/4/20230476.

[8] Evans, B., Faul, A., Fleming, A., Vaughan, D.G., & Hosking, J.S. (2023). Unsupervised machine learning detection of iceberg populations within sea ice from dual-polarisation SAR imagery. *Remote Sensing of Environment*, 297, article number 113780. doi: 10.1016/j.rse.2023.113780.

[9] Filho, J.F.M.R., & Bélanger, P. (2021). Probe standoff optimization method for phased array ultrasonic TFM imaging of curved parts. *Sensors*, 21(19), article number 6665. doi: 10.3390/s21196665.

[10] Gavrylenko, S., & Chelfk, V. (2023). Development of method base on fuzzy decision trees for identification of the computer systems state. *Navigation and Communication Systems*, 1(71), 78-83. doi: 10.26906/sunz.2023.1.078.

[11] Hochstuhl, S., Pfeffer, N., Thiele, A., Hammer, H., & Hinz, S. (2023). Your input matters – comparing real-valued PolSAR data representations for CNN-based segmentation. *Remote Sensing*, 15(24), article number 5738. doi: 10.3390/rs15245738.

[12] Huang, Z. (2024). A research on image recognition and classification based on traditional machine learning and deep learning. *Transactions on Computer Science and Intelligent Systems Research*, 5, 766-773. doi: 10.62051/0dbqaa10.

[13] Imad, H., Sara, Z., Hajji, M., Yassine, T., & Abdelkrim, N. (2024). Recent advances in SAR image analysis using deep learning approaches: Examples of speckle denoising and change detection. In B. Benhala, A. Raihani & M. Qbadou (Eds.), *4th international conference on innovative research in applied science, egineering and technology* (pp. 1-6). Morocco: Institute of Electrical and Electronics Engineers. doi: 10.1109/IRASET60544.2024.10549456.

[14] Iqbal, J., Vogt, M., & Bajorath, J. (2020). Activity landscape image analysis using convolutional neural networks. *Journal of Cheminformatics*, 12, article number 34. doi: 10.1186/s13321-020-00436-5.

[15] Kanmani, K., Padmanabhan, V., & Pari, P. (2023). Accuracy assessment of different classifiers for sustainable development in landuse and landcover mapping using sentinel SAR and landsat-8 data. *EAI Endorsed Transactions on Energy Web*, 10. doi: 10.4108/ew.4141.
[16] Konishi, B., Hirose, A., & Natsuaki, R. (2021). Complex-valued reservoir computing for interferometric SAR applications with low computational cost and high resolution. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 7981-7993. doi: 10.1109/jstars.2021.3102620.
[17] Kruk, R., Fuller, M.C., Komarov, A.S., Isleifson, D., & Jeffrey, I. (2020). Proof of concept for sea ice stage of development classification using deep learning. *Remote Sensing*, 12(15), article number 2486. doi: 10.3390/rs12152486.
[18] Li, Q., Bai, X., Hu, L., Li, L., Bao, Y., Geng, X., & Yan, X.-H. (2024). SAR image semantic segmentation of typical oceanic and atmospheric phenomena. *Earth System Science Data*. doi: 10.5194/essd-2024-222.
[19] Li, X., Wang, Z., Li, J., & Mu, P. (2023). Swin-transformer based target detection with enhanced maritime SAR images data. In S. Xu & S. Sazena (Eds.), *International conference on signal processing and communication technology (SPCT 2022)*. Harbin: Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.2673920.
[20] Li, Y., Zhang, S., Li, X., & Ye, F. (2023). Remote sensing image classification with few labeled data using semisupervised learning. *Wireless Communications and Mobile Computing*, 2023(1), article number 7724264. doi: 10.1155/2023/7724264.
[21] Liu, S., Pu, N., Cao, J., & Zhang, K. (2022). Synthetic aperture radar image despeckling based on multi-weighted sparse coding. *Entropy*, 24(1), article number 96. doi: 10.3390/e24010096.
[22] Liu, T., Li, Y., Cao, Y., & Shen, Q. (2017). Change detection in multitemporal synthetic aperture radar images using dual-channel convolutional neural network. *Journal of Applied Remote Sensing*, 11(4), article number 042615. doi: 10.1117/1.jrs.11.042615.
[23] Majji, S.R., Chalumuri, A., Kune, R., & Manoj, B.S. (2022). Quantum processing in fusion of SAR and optical images for deep learning: A data-centric approach. *IEEE Access*, 10, 73743-73757. doi: 10.1109/access.2022.3189474.
[24] Meng, H., Li, C., Liu, Y., Gong, Y., He, W., & Zou, M. (2023). Corn land extraction based on integrating optical and SAR remote sensing images. *Land*, 12(2), article number 398. doi: 10.3390/land12020398.
[25] Monsalve-Tellez, J.M., Torres-León, J.L., & Garcés-Gómez, Y.A. (2022). Evaluation of SAR and optical image fusion methods in oil palm crop cover classification using the random forest algorithm. *Agriculture*, 12(7), article number 955. doi: 10.3390/agriculture12070955.
[26] Nillmani, N., Sharma, N., Saba, L., Khanna, N., Kalra, M., Fouda, M., & Suri, J. (2022). Segmentation-based classification deep learning model embedded with explainable AI for COVID-19 detection in chest X-ray scans. *Diagnostics*, 12(9), article number 2132. doi: 10.3390/diagnostics12092132.
[27] Oumarou, H., & Rismayanti, N. (2024). Automated classification of empon plants: A comparative study using Hu Moments and K-NN algorithm. *Indonesian Journal of Data and Science*, 4(3), 206-214. doi: 10.56705/ijodas.v4i3.115.
[28] Pan, J., Zhang, H., Wu, W., Gao, Z., & Wu, W. (2022). Multi-domain integrative Swin transformer network for sparse-view tomographic reconstruction. *Patterns*, 3(6), article number 100498. doi: 10.1016/j.patter.2022.100498.
[29] Poplavskyi, O. (2024). Information technology for image data processing based on hybrid neural networks using geometric features. *Information Technologies and Computer Engineering*, 21(2), 4-16. doi: org/10.31649/1999-9941-2024-60-2-4-16.
[30] Schmitt, M., Hughes, L.H., & Zhu, X.X. (2018). The SEN1-2 dataset for deep learning in SAR-optical data fusion. *ISPRS Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences*, 4(1), 141-146. doi: 10.5194/isprs-annals-iv-1-141-2018.
[31] Villarroya-Carpio, A., & López-Sánchez, J.M. (2023). Multi-annual evaluation of time series of Sentinel-1 interferometric coherence as a tool for crop monitoring. *Sensors*, 23(4), article number 1833. doi: 10.3390/s23041833.
[32] Vu, V.T., Pettersson, M.I., Palm, B.G., Alves, D.I., & Gomes, N.R. (2021). Changing flight heading during pass to enhance SAR change detection performance. *IET Radar Sonar & Navigation,* 15(8), 817-826. doi: 10.1049/rsn2.12058.
[33] Wang, L., Jin, G., Xiong, X., Zhang, H., & Wu, K. (2022). Object-based automatic mapping of winter wheat based on temporal phenology patterns derived from multitemporal Sentinel-1 and Sentinel-2 imagery. *ISPRS International Journal of Geo-Information*, 11(8), article number 424. doi: 10.3390/ijgi11080424.
[34] Wei, B., Huang, M., Zhang, Y., Xu, Y., Liu, X., & Xiang, X. (2021). Boosting ship detection in SAR images with complementary pretraining techniques. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 8941-8954. doi: 10.1109/jstars.2021.3109002.

[35] Wu, Y., Xia, Y., Jiang, M., Li, S., Chen, M., Zhao, Y., & Teng, D. (2023). Multisource remote sensing image registration based on geometric constraints. In C. Zuo (Ed.), *International conference on remote sensing, surveying, and mapping (RSSM 2023)* (article number 127100J). Changsha: Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.2682662.
[36] Xue, Z., & Zhang, M. (2020). Multiview low-rank hybrid dilated network for SAR target recognition using limited training samples. *IEEE Access*, 8, 227847-227856. doi: 10.1109/access.2020.3046274.
[37] Zeng, Z., Tan, X., Chen, Z., Huang, Y., Tang, S., & Wan, J. (2022). Robust image similarity metric method for SAR images. *Electronics Letters*, 58(13), 508-510. doi: 10.1049/ell2.12516.
[38] Zengguo, S., Zhao, M., & Jia, B. (2021). A GF-3 SAR image dataset of road segmentation. *Information Technology and Control*, 50(1), 89-101. doi: 10.5755/j01.itc.50.1.27987.
[39] Zhang, H., Li, G., & Lin, H. (2016). An automatic co-registration approach for optical and SAR data in urban areas. *Annals of GIS*, 22(3), 235-243. doi: 10.1080/19475683.2016.1199595.
[40] Zhang, R., Tang, X., You, S., Duan, K., Xiang, H., & Luo, H. (2020). A novel feature-level fusion framework using optical and SAR remote sensing images for land use/land cover (LULC) classification in cloudy mountainous area. *Applied Sciences*, 10(8), article number 2928. doi: 10.3390/app10082928.
[41] Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7), 13001-13008. doi: 10.1609/aaai.v34i07.7000.

Машинне навчання та глибинне навчання для класифікації SAR-зображень

Юрій Бровка
Аспірант
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0009-0004-3708-9747

Анотація. У дослідженні проведено комплексний аналіз сучасних методів машинного та глибинного навчання для класифікації радіолокаційних зображень із синтезованою апертурою. Метою роботи було визначення архітектур і підходів, які забезпечують високу точність класифікації при збереженні ефективності обчислювальних ресурсів. Основну увагу приділено вирішенню ключових викликів: спекл-шуму, геометричних спотворень та обмеженості розмічених даних. Методологія дослідження включала систематичний аналіз наукової літератури за 2015-2024 роки та вивчення поляризаційних характеристик Synthetic Aperture Radar-зображень із використанням платформи Copernicus Browser. Було досліджено ефективність традиційних методів машинного навчання (Support Vector Machines, Random Forest) та сучасних архітектур глибокого навчання (ResNet, U-Net, Vision Transformer). Особливу увагу приділено впливу адаптивної фільтрації спекл-шуму за допомогою фільтра Lee при різних розмірах вікна (3×3; 5×5; 7×7) на якість класифікації. Результати показали, що глибокі нейронні мережі мають переваги перед традиційними методами завдяки здатності автоматично виділяти ієрархічні представлення ознак. ResNet забезпечує високу точність класифікації, U-Net ефективна для сегментації, а Vision Transformer враховує глобальні залежності. Оптимальний баланс між придушенням спекл-шуму та збереженням деталей досягається при використанні фільтра Lee з розміром вікна 5×5. Однією з проблем залишається обмеженість розмічених даних. Для її вирішення розглянуто напівконтрольоване навчання, що покращує нормалізацію ознак і продуктивність моделей. Перспективним напрямком є використання комплекснозначних нейронних мереж для оптимізації обчислювальних витрат. Результати мають практичне значення для автоматизованої класифікації Synthetic Aperture Radar-зображень у задачах екологічного моніторингу, оцінки стану сільськогосподарських угідь та дистанційного зондування

Ключові слова: нейронні мережі; комплекснозначні обчислення; поляризаційні характеристики; радіолокаційне зондування; спекл-фільтрація

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 30 No. 1. 2025

UDC 004.056.55:004.75
DOI: 10.62660/bcstu/1.2025.21

Sonar image processing for improved underwater environment modelling

Oleksandr Katrusha*
Master
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Beresteiskyi Ave., Kyiv, Ukraine
https://orcid.org/0009-0008-7097-4843

Artem Burdeinyi
Master
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Beresteiskyi Ave., Kyiv, Ukraine
https://orcid.org/0009-0009-3113-2149

Abstract. The purpose of the study was to present a sequence of development of an algorithm for improving the quality of images obtained using side-scan sonar. Mathematical image processing techniques such as contrast enhancement, edge processing, and colorimetric techniques were used to achieve this goal. Steps to improve image contrast included normalising signal intensity, adaptive contrast enhancement using limited contrast alignment of histograms, and correction of uneven lighting. The sonar radiation pattern and image intensity normalisation scheme were demonstrated. The contrast limited adaptive histogram equalisation filter showed higher values of the peak signal-to-noise ratio and structural similarity index compared to conventional histogram alignment, indicating better preservation of detail, image structure, and noise reduction. Analysis of edge processing, in particular by Canny and Sobel, has shown their potential effectiveness in improving the detail of underwater structures. In addition, the use of Gaussian smoothing allowed reducing the level of high-frequency noise and make textures smoother. As a result, there was a decrease in graininess, softness of object contours, and overall smoothing of the scene. In addition, cubic spline regression showed normalised image data. In turn, colorimetric analysis focused on converting images between greyscale and colour spaces, which made it easier to identify underwater objects and structures. An example of Hue-Saturation-Value components was given, which demonstrated different effects on the quality of sonar image visualisation. The Value component provided the most expressive distinction between the object and the background, while the Hue component was ineffective for structure analysis. The combination of Value and Saturation allowed for improved contour detail. Optimisation of the pseudo-colour gamut allowed adapting the image to different tasks, contributing to more accurate object recognition. The results obtained confirm the feasibility of using the presented methods in a wide range of applied tasks related to visualisation and analysis of underwater environments

Keywords: contrast adjustment; edge selection; colourimetric analysis; distortion elimination; adaptive anti-aliasing

Article's History: Received: 18.09.2024; Revised: 20.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Katrusha, O., & Burdeinyi, A. (2025). Sonar image processing for improved underwater environment modelling. *Bulletin of Cherkasy State Technological University*, 30(1), 21-34. doi: 10.62660/bcstu/1.2025.21.

*Corresponding author

INTRODUCTION

Sonar is one of the main technologies for studying the underwater environment, in particular, for detecting and mapping objects on the sea floor. This is the process of using sound waves to determine the location and characteristics of underwater objects, where sonar is the main element. A sonar is a device that sends and receives acoustic signals, and a sonogram is an image that is obtained as a result of sonar operation. One of the most common types of sonar is side-scan sonar (SSS), which is used to create detailed images of the underwater surface by sequentially scanning it using narrow-band ultrasonic waves. The SSS system operates on the principle of emitting sound waves reflected from objects on the sea floor and receiving reflected signals, which allows creating a clear image with high resolution. The use of this type of sonar is important in research where accurate visualisation of underwater structures and objects is required. One of the main problems in using sonar is the relatively poor quality of the resulting images, often due to factors such as poor contrast, noise, and distortion that make it difficult to interpret the data. A significant obstacle is that the appearance of band noise from roll variations negatively affects image clarity, particularly making object recognition and seabed segmentation difficult (Katrusha, 2024). In addition, existing image processing methods do not always effectively cope with the correction of uneven lighting and image detail, which creates difficulties in accurately detecting underwater objects and structures.

For a better understanding of this topic, it is necessary to consider other papers. For example, V.I. Ivaniuk *et al.* (2024) proposed the integration of colour reconstruction and deep learning (DL) on board autonomous unmanned underwater vehicles to improve the accuracy and speed of underwater object detection, considering the complexity of the environment. For its part, N.F. Bogomolov *et al.* (2021) analysed speckle interferogram and computer-generated optical image processing based on a model in Matlab. The results showed the effectiveness of an approach for optical diagnostics of biological micro-objects using cross-correlation analysis. Moreover, O. Klein *et al.* (2023) developed the architecture of a cyberphysical computer vision system for detecting anomalies in images, especially for objects in motion. They proposed a hybrid approach that involves centralised management of the system with the ability to scale the level of autonomy for each task. Sonar devices use different frequencies-from 6.5 kHz to 1 MHz, which provides a sonogram resolution of 60 m to 1 cm. T. Fletcher *et al.* (2024) found that recreational sonars, such as Lowrance, provide sufficient resolution and accuracy in classifying environments at a lower cost and with less data processing time compared to professional systems, making them an effective tool for mapping littoral areas of water bodies. Y. Zhang *et al.* (2021) analysed five image enhancement algorithms: histogram equalisation (HE), greyscale stretching (GS), mean filter (MF), Multi Scale Retinex (MSR), and wavelet transform (WT), and HE, GS, and MSR provided the best results. However, traditional optical computer vision techniques such as MSR inspired S. Li *et al.* (2022) and Y. Liu & X. Ye (2023) to develop specific enhancement methods for sonar imaging. Y. Liu & X. Ye presented a greyscale correction method for SSS images with complex terrain, based on acoustic attenuation compensation and improving the bottom relief display and mosaic quality of SSS images. S. Li *et al.* proposed a new method of radiometric correction that considers the features of SSS image formation and uses low-angle and anisotropic variational constraints for lighting and albedo components, effectively solving the distortion problem.

Regarding the evaluation of the results of methods, it can be noted that there is no universal approach to evaluation. V.D. Nguyen *et al.* (2023) provided an empirical method for estimating the radiation pattern for processed images based on acoustic transducers using geometric backscattering model for SSS. Advances in machine learning (ML) and artificial intelligence (AI) should be considered, which open up new opportunities for improving intensity correction, in particular, for automating the identification and correction of complex distortion patterns. For example, K. Sivachandra & R. Kumudham (2024) conducted a review of research on object detection and classification in SSS images using deep and ML techniques, and image processing, analysing the use of convolutional neural network (CNN), data sets, detection accuracy, and classification. Similarly, Y. Steiniger *et al.* (2022) provided an overview of studies using DL for sonar image analysis, in particular, for feature detection, classification, detection, and image segmentation of SSS synthetic aperture sonars, with a focus on CNN, including the creation of an open dataset and comparative studies of DL methods. Such approaches are widely used and popular in sonar image processing and computer vision technologies. In the considered studies, there is no comprehensive algorithm for improving the quality of sonar images, which would include simultaneously improving contrast, edge processing, and colour interpretation. The purpose of the study was to develop an algorithm that combines these methods to improve the information content and accuracy of sonar images. The task was to analyse improved image quality, apply edge processing techniques to improve sonar image detail, and explore colorimetric techniques to improve image interpretation.

MATERIALS AND METHODS

This study consisted of three subsections that considered methods to improve image contrast, edge processing to improve detail, and colorimetric methods to simplify data interpretation. In the first section, which was devoted to improving image contrast, an example of a sonar radiation pattern was given, where for port, the main lobe had a frequency of 32.15 kHz (kilohertz), and

the first side lobe – 30.37 kHz, while for starboard, the first side lobe corresponded to a frequency of 32.15 kHz, and the main lobe – 30.37 kHz. The importance of the SSS radiation pattern was to explain possible variations in signal intensity that affect the contrast and uniformity of illumination on the sonogram. Since uneven exposure can make visual interpretation difficult, consideration of these features helped to better assess the sources of reduced image quality. The method for improving image contrast included the following steps: intensity normalisation, adaptive contrast enhancement, and correction of uneven lighting. Normalisation of intensity took into account the features of the sonar signal, which was reflected in the scheme of the stage of normalisation of the intensity of the sonar image. The contrast limited adaptive histogram equalisation (CLAHE) approach was considered for adaptive contrast improvement. To evaluate its effectiveness, an example of SSS image processing using HE and CLAHE filters was provided. The quality of processed images was evaluated using metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and mean squared error (MSE). The last step was the correction of uneven lighting, where examples of eliminating striped noise and excessive illumination were considered. All three stages are summarised in a Table that contains the purpose of each stage, its description, and usage examples.

For its part, the second section of the study focused on edge processing to improve detail. In the first stage (image pre-processing), the application of cubic spline regression was analysed, which allowed the data to be normalised. This example showed data normalisation with the original image before image processing and the result after normalisation and filtering were applied. The results of using Gaussian blur (GB) to eliminate noise were also demonstrated. For this purpose, a sonar image of the sunken ship obtained using SSS was used. Image processing was performed in the Pixlr Editor Image Editor, where GB with a radius of 15 was applied. In general, the edge selection step was performed using techniques such as Canny and Sobel, which were compared by their advantages and limitations. Morphological operations (dilation, erosion) and high-pass filters (Laplace, Gauss) were considered for the edge amplification stage. In addition, a diagram of this stage was developed, demonstrating the choice of optimal approaches depending on the type of image. Similar to the first subsection, a Table was created to summarise the results with a description of all the stages of the subsection, their purpose, and examples.

In turn, the third section focused on applying colorimetric techniques to improve image interpretation. This included the following steps: converting images between greyscale and colour spaces, using colorimetric models to analyse structures, and optimising the pseudo-colour gamut. For the first stage, an example of the Hue, Saturation, and Value (HSV) colour model components was given, which helped to more accurately identify underwater objects. Using the GNU Image Manipulation Programme (GIMP), the underwater vessel image was divided into separate HSV components to assess their impact on contrast and detail. The application of colorimetric models was described in the form of a diagram that demonstrates the analysis of structures and details. Pseudo-colour gamut optimisation was aimed at adapting images to specific tasks and simplifying their interpretation. All stages of this division were also summarised in the corresponding table, which contained a description, goal, and examples of results.

RESULTS

Contrast enhancement methods for the image processing quality improvement algorithm

Sonar image processing is a key method in modelling the underwater environment. Based on SSS capabilities, it is possible to create detailed maps of the sea floor, identify objects on its surface, and analyse ecosystems. However, there is a need to develop an algorithm for improving the quality of image processing obtained using sonar, using modern mathematical methods. Contrast enhancement should be highlighted as one of the methods of such an algorithm, since low contrast is one of the main obstacles to accurate analysis of underwater structures and objects. This problem is caused by several factors, in particular, the attenuation of the signal intensity due to the distance of the object from the radiation source, the influence of the sonar radiation pattern (Fig. 1), and uneven lighting near the nadir zone. All of these factors create variations in intensity that make it difficult to automatically detect and identify objects.

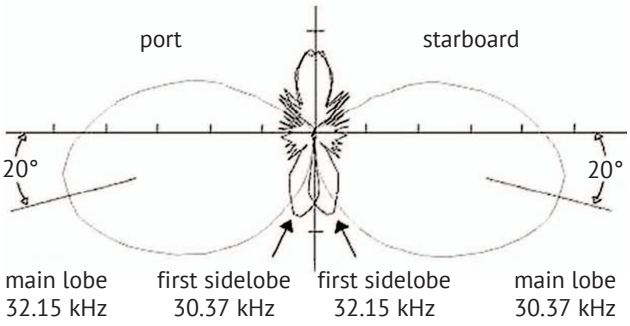

Figure 1. Example of a sonar radiation pattern
Source: compiled by the authors based on P. Blondel (2009)

The sonar radiation pattern causes changes in intensity on the sonogram, which can cause uneven lighting and striped noise. In addition, steps to improve the contrast of sonar images should be considered, considering the specifics of sonogram processing and typical distortions characteristic of this type of data. The first step is to normalise the intensity, which eliminates global lighting variations and equalises the brightness by scaling the pixel values to the specified range (Fig. 2). This allows preparing the data for further processing.

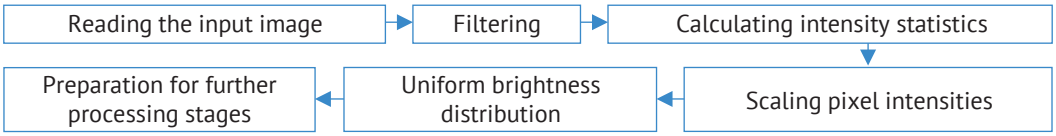

Figure 2. Diagram of the sonar image intensity normalisation stage
Source: compiled by the authors

After normalising the intensity, the next step is to apply an adaptive approach to contrast enhancement, such as CLAHE (Fig. 3). This step is an improvement on conventional histogram alignment, as it works locally within small blocks of the image, while maintaining the naturalness of textures and preventing excessive noise amplification. CLAHE also uses a limit on the contrast level, which allows avoiding overexposure in certain areas.

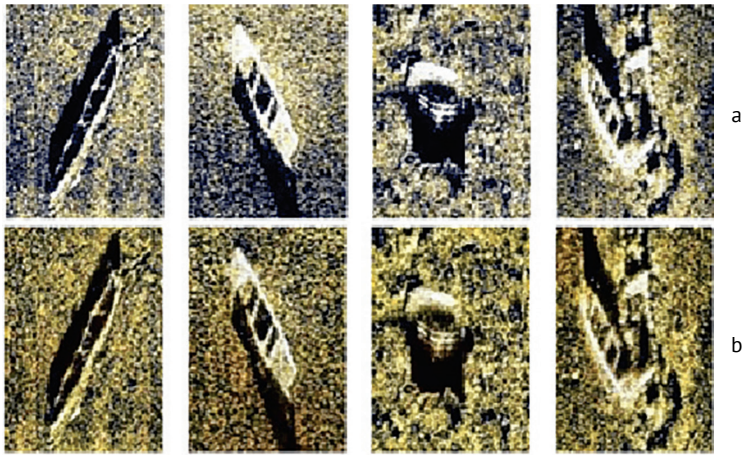

Figure 3. Example of using the usual HE (a) and CLAHE (b) for SSS images
Source: compiled by the authors

In Figure 3 (b), the contrast with CLAHE is noticeably more evenly improved compared to HE (a), without the excessive noise gain observed in HE. CLAHE preserves the details of objects better, especially in shady areas and on borders, providing a more natural look to the image. The analysis shows that CLAHE performs better than HE in most image quality indicators (Table 1).

Table 1. Metrics for evaluating image quality

Image	Method	Entropy	PSNR	SSIM	MSE
Image 1	HE	7.847	12.943	0.698	3,301.756
	CLAHE	7.767	16.345	0.779	1,508.438
Image 2	HE	7.945	12.613	0.697	3,562.527
	CLAHE	7.968	15.893	0.764	1,674.00
Image 3	HE	7.953	11.237	0.610	4,890.696
	CLAHE	7.915	15.479	0.713	1,841.199
Image 4	HE	7.939	11.639	0.637	4,457.708
	CLAHE	7.949	15.670	0.733	1,762.108

Source: compiled by the authors based on Z. Lu et al. (2023)

The results of metrics (PSNR, SSIM, MSE) confirmed the superiority of CLAHE over HE in preserving structural information and reducing distortion. Overall, CLAHE provides a more balanced quality improvement, reducing noise and artefacts while preserving details and structures in sonar images. At the final stage of sonar image processing, uneven lighting is corrected, which is one of the main problems in sonography. This step is based on the analysis of intensity histograms for each local area of the image. First, a separate histogram is created for each local area (subzone of the image), which reflects the distribution of pixel intensity in this part of the image. Then, based on this histogram, the average intensity value for each region is determined. To compensate for uneven lighting, the approach of adapting local intensity to global image parameters is used. This means that the average intensity value of each local area is adjusted so that it corresponds to

the average or specified global intensity level of the entire image. Thus, local brightness variations that can be caused by lighting distortions are reduced, in particular, in the nadir zone. It is necessary to demonstrate an example of excessive illumination in such an area, where bright areas in the centre of the image are detected, and striped noise in the form of horizontal lines of varying intensity and length (Fig. 4). This example is a typical problem to consider when correcting sonar images. In general, the analysed stages of image processing allow achieving improvements in the accuracy and clarity of images, which increases the efficiency of further analysis of underwater objects. Table 2 summarises these steps, their goals, and application examples.

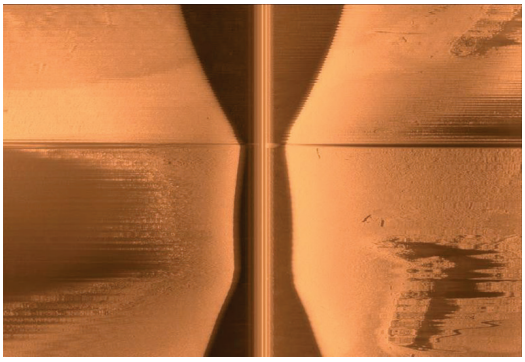

Figure 4. Example of excessive light and striped noise
Source: compiled by the authors

Table 2. Steps to improve the contrast of sonar images

Processing stage	Stage goal	Examples
Normalisation of intensity	Eliminate global lighting variations and equalise brightness across the entire image plane	Eliminate dark spots or bright areas that occur due to uneven lighting in the sonar image
CLAHE filter	Improve local contrast without increasing noise and distortion	Improve contrast in an area with low visibility against the background, for example, increase the clarity of small objects on the sea floor without oversaturating the brightness
Correction of uneven lighting	Reduction of local brightness variations caused by lighting distortions, in particular, in the nadir zone	Compensation for excessive illumination in the central part of the image, where intense signal reflection is observed, or correction of striped noise in the form of horizontal lines

Source: compiled by the authors

Therefore, using the considered steps to improve contrast allows reducing the impact of global and local brightness variations, reduce noise levels, and increase image detail. Metric analysis (PSNR, SSIM, MSE) confirmed that CLAHE provides better image quality compared to conventional histogram alignment, and correction of uneven lighting helps eliminate artefacts in the nadir zone. Thus, the proposed approach improves the quality of sonograms, which increases the accuracy of their subsequent analysis.

Overview of edge processing steps to improve image detail

Detailed sonar images are an important aspect for ensuring accurate analysis of underwater objects and structures. Problems associated with distortion (blurred contours, reduced contrast, and abnormal reflections) or blurring can significantly complicate the process of automatically detecting and classifying objects on the sea floor. To eliminate these problems, edge processing should be used, which is based on image preprocessing, edge detection and edge enhancement. The main task of the preprocessing stage is to reduce noise, in particular, signal reflection, electronic interference, and high-frequency components that degrade the detail of objects. To illustrate this process, an example of data normalisation is given, which shows the original image with noise and distortion, and the result after normalisation and filtering, showing a uniform distribution of intensity and noise reduction (Fig. 5).

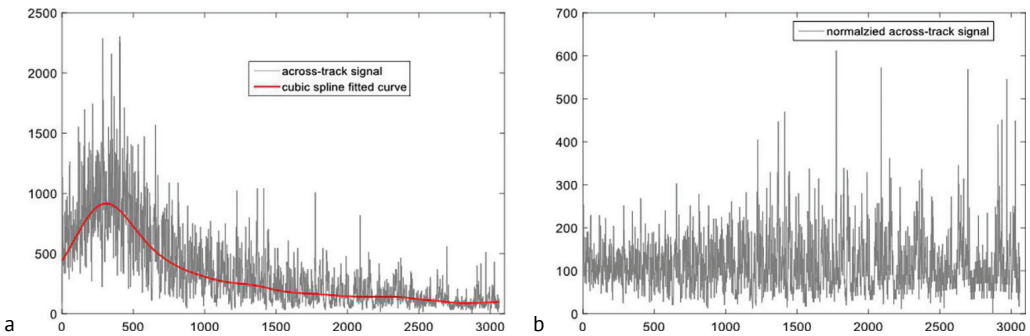

Figure 5. Example of data preprocessing by cubic spline regression
Notes: a – original image before processing; b – result after normalisation and filtering are applied
Source: compiled by the authors based on M. Al-Rawi *et al.* (2017)

This step applies filters to smooth out images and reduce random interference. One effective approach is to use GB to filter images. It is based on convolution of the image with a Gaussian core, which suppresses high-frequency interference, in particular, small chaotic details that look like noise. It is worth demonstrating a sonar image of the sunken ship obtained using the Side Scan Sonar survey (n.d.), which was processed in the Pixlr Editor (n.d.) (Fig. 6).

As a result of applying GB with a radius of 15, a noticeable reduction in grain size was achieved, which made the textures smoother (Fig. 7). The edges of objects have become softer, which reduces the likelihood of false detections in subsequent processing, but at the same time can lead to the loss of some parts. The overall appearance of the scene has become more smoothed out, which can be useful for further segmentation or automatic detection of underwater structures.

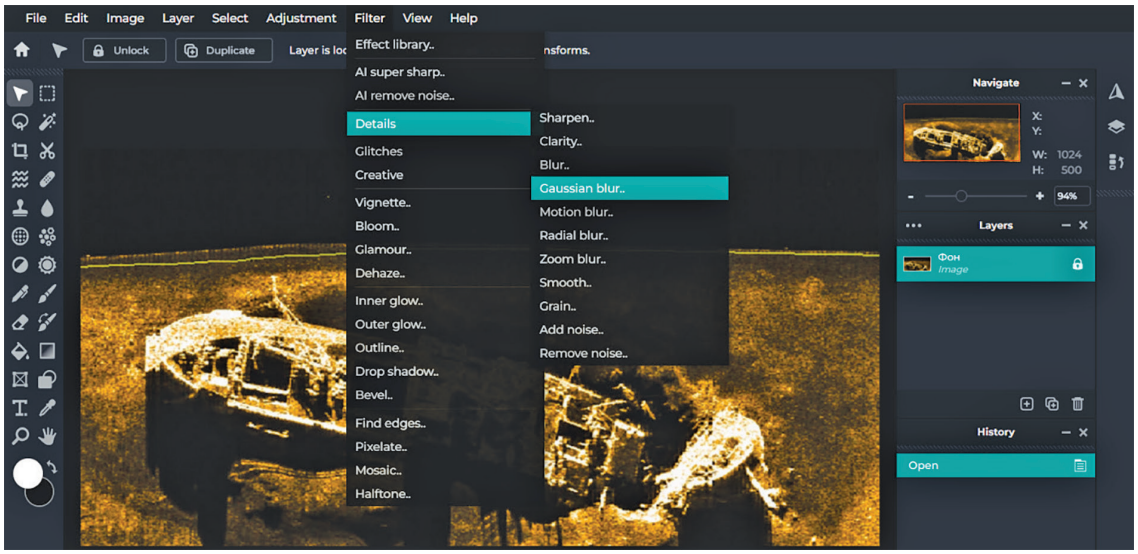

Figure 6. Configuring parameters for GB image processing
Source: compiled by the authors based on Side Scan Sonar survey (n.d.), Pixlr Editor (n.d.)

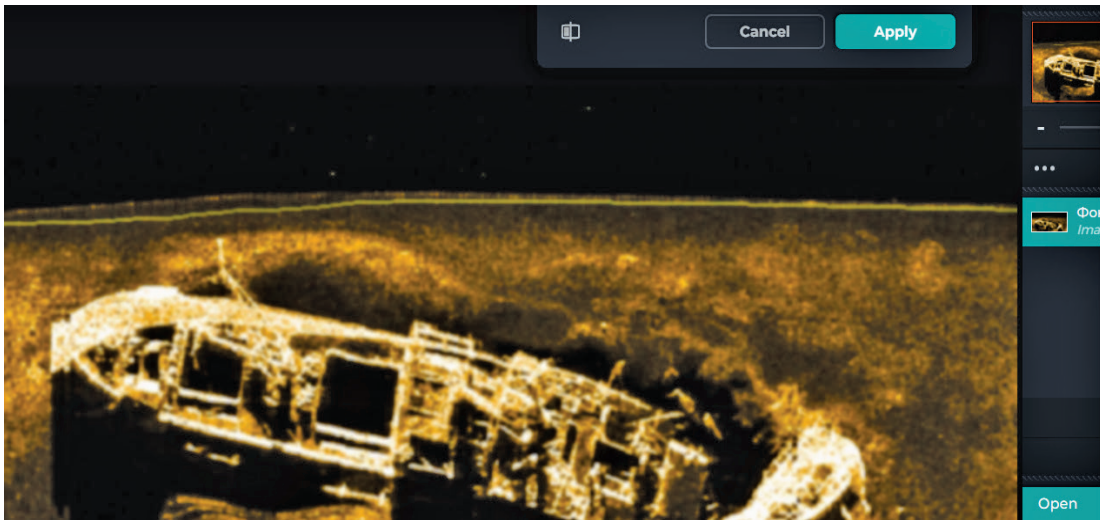

Figure 7. Result of using GB
Source: compiled by the authors based on Side Scan Sonar survey (n.d.), Pixlr Editor (n.d.)

Therefore, GB is particularly effective in cases where the original image has uneven illumination or contains abnormal reflections. In general, this approach is basic for ensuring the accuracy of edge detectors, as it reduces the likelihood of false detections caused by random noise or intensity anomalies. After pre-filtering noise, the main task is to highlight the edges that form the contours of underwater objects. This step aims to identify dramatic changes in pixel intensity corresponding to boundaries between different structures or materials on the sea floor. For this purpose, specialised approaches based on gradient analysis are used, in particular, Canny and Sobel (Table 3).

Table 3. Comparison of Canny and Sobel

Approach	Advantages	Disadvantages
Canny	High accuracy of edge detection	Higher computational complexity compared to Sobel
	Minimisation of false positives by suppressing non-maximal pixels	Sensitivity to parameters (thresholds)
	Ability to highlight even weak edges if they border on strong ones	It is not always optimal for real time due to the long processing time of large images
Sobel	Easy implementation and high speed of operation	Lower edge detection accuracy compared to Canny
	Effective for rough edge detection and initial processing	Vulnerable to noise and requires additional pre-anti-aliasing
	Used for tasks that do not require high-level detail	Poorly distinguishes between weak edges or details in low-contrast images

Source: compiled by the authors

Canny technology ensures high accuracy of edge selection and minimises false positives. The algorithm works sequentially: calculating the intensity gradient, suppressing non-maximal pixels, and processing thresholds that connect weak contours with strong ones. Instead, Sobel is less computationally complex and is used to quickly estimate gradients, although it is less accurate and noise-sensitive. That is, Canny is suiTable for detailed analysis, and Sobel is suiTable for pre-processing or when speed is important. An equally important step is to strengthen the edges, which allows to get clearer borders of objects. This is necessary for further tasks, in particular, classification and detection of underwater structures. The main method of reinforcement is the use of morphological operations, such as dilation and erosion. Dilation increases the thickness of the edges, filling in the gaps between them, which allows increasing their expressiveness. Erosion, on the other hand, reduces the thickness of the edges, eliminating minor or indistinct contours. It is important that these approaches do not change the main contours of objects, but only improve their reflection in the image.

Another way to amplify edges is to use high-pass filters, such as Laplace filters or Gaussian operations. Such filters emit dramatic changes in intensity, strengthening the boundaries between different materials or structures in the image. They make the boundaries of objects clearer and more expressive for further analysis, and the image becomes more informative and allows more accurately determining the location of underwater objects. This stage is crucial for further automated detection and classification, where it is important to have clear and precise contours that correspond to the actual characteristics of underwater objects. The diagram in Figure 8 illustrates the process of edge amplification, from choosing how to amplify edges to filters and getting well-defined edges ready for further analysis.

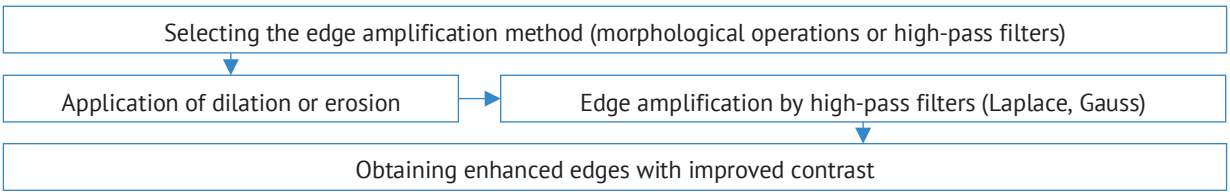

Figure 8. Diagram of the edge reinforcement stage

Source: compiled by the authors

Image pre-processing, highlighting, and edge enhancement are critical to improving the accuracy and clarity of sonar images. They allow significantly reducing the impact of noise, clearly highlighting the contours of underwater objects, and increasing their expressiveness for further analysis (Table 4).

Table 4. Steps for processing the edges of sonar images

Processing stage	Stage goal	Examples
Pre-processing	Reduce the impact of noise and distortion due to various physical and technical factors	Elimination of signal reflections, electronic interference, reduction of high-frequency noise from the image
Edge selection	Identification of sudden changes in pixel intensity for finding object boundaries	Detecting the contours of underwater objects (such as corals or sunken structures) using Canny or Sobel
Edge reinforcement	Improve clear edge expressiveness for further analysis and classification	Application of morphological operations (dilation, erosion), high-pass filters (Laplace, Gauss)

Source: compiled by the authors

That is, pre-processing by noise filtering helps to eliminate interference, edge selection ensures accurate delineation of the contours of underwater objects, and edge amplification by morphological operations and high-pass filters increases their expressiveness for further classification. These steps form a qualitative basis for automated analysis of underwater structures, minimising false detection, and loss of important details.

Analysis of colorimetric methods to improve image interpretation

Colorimetric sonar image processing techniques improve the recognition of underwater structures, which contributes to their more accurate identification. The first stage of improving the following images involves converting images between greyscale and colour spaces depending on the task at hand. This can include either translating the grey image into a colour space to improve visualisation, or decomposing the colour image into separate components for further analysis. During this transformation, pixel values are scaled according to the selected colour palette or individual colour components are extracted, which provides a basis for a deeper interpretation of the object structure. For example, the HSV space (Fig. 9) should be used because of its ability to separate components of brightness, colour tone, and saturation, which facilitates visual interpretation. At this stage, the pixel values in the greyscale are scaled and distributed according to the colour palette, which creates the basis for further analysis.

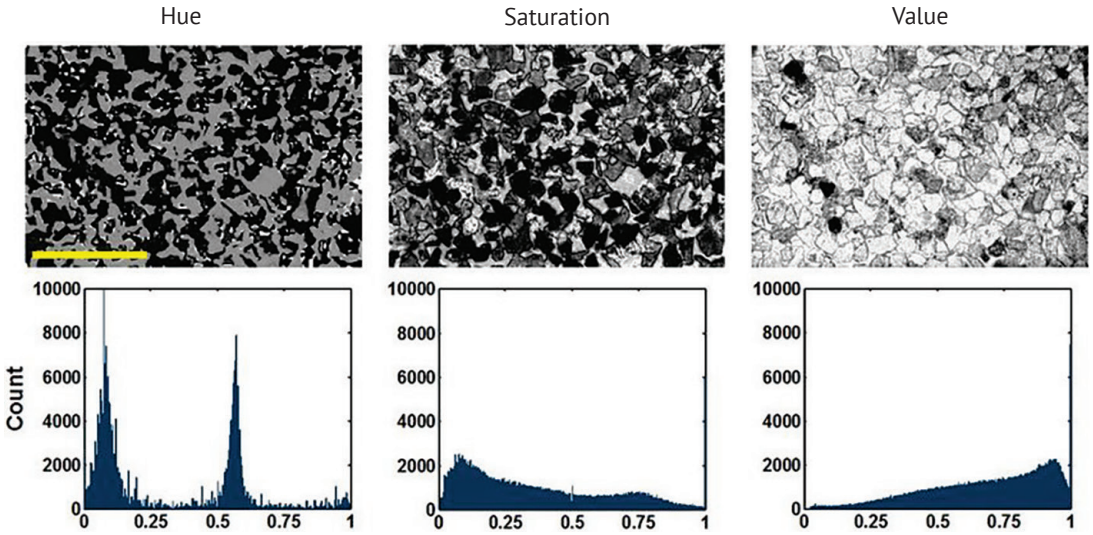

Figure 9. Example of HSV components

Source: compiled by the authors based on R. Richa (2010)

To test the effectiveness of using the HSV colour space, the method of identifying individual components of this space on the sonar image of a sunken ship (Side Scan Sonar survey, n.d.). This image contains a yellow-black palette typical of many sonar systems. The main contours of the vessel are clearly visible, but the contrast of some details is insufficient, which makes it difficult to identify them. The analysis was performed using the GIMP software environment, which allows separating colour channels and evaluating their impact on object visualisation (Fig. 10).

For further analysis, three key components of HSV were identified: Hue, Saturation, and Value (Brightness) (Fig. 11). The Hue component shows a low contrast between the object and the environment. This is because the hue of the colour in the original image is fairly uniform. In this case, the Hue component does not contain enough information for high-quality detail recognition, since hue changes are not the main factor in distinguishing structures in the sonar image. For its part, the Value component shows a significant increase in contrast. The main outlines of the ship have become more expressive, which helps to better identify its shape and structure. This component turned out to be one of the most informative, since it is directly related to the level of reflected sonar signal. Objects with high brightness are more reflective, which allows highlighting important features. Moreover, the result of the Saturation component is characterised by even higher contrast, but it has certain disadvantages. Some fine details are lost, and the overall texture of the image becomes more simplified. This can be useful in cases where it is necessary to clearly highlight the main outline of an object, but to preserve finer details, it is advisable to combine this component with brightness.

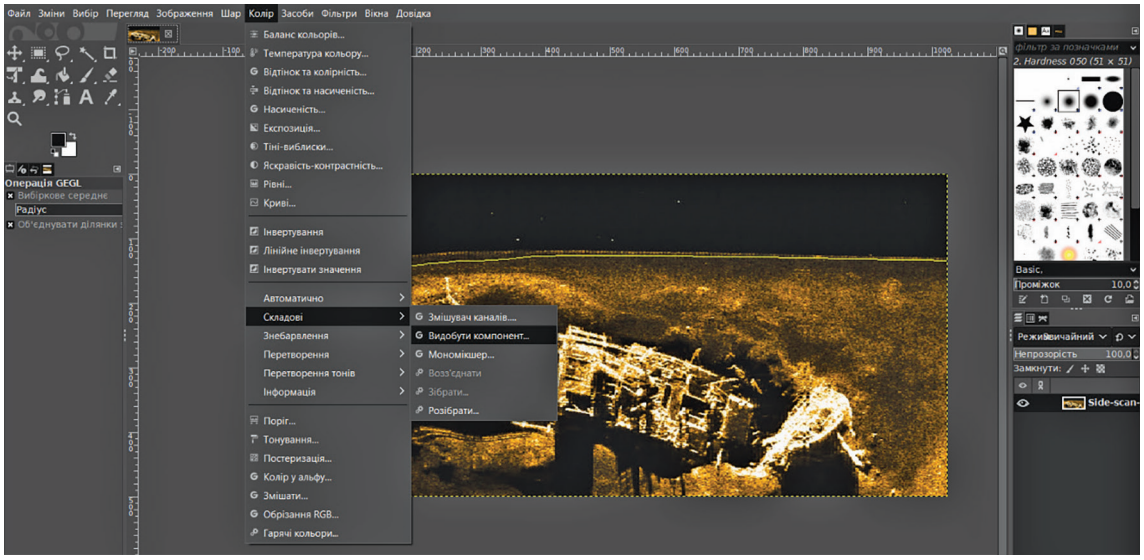

Figure 10. Configuring settings in GIMP

Source: compiled by the authors based on Side Scan Sonar survey (n.d), GIMP (n.d.)

Figure 11. Application of HSV components

Source: compiled by the authors based on Side Scan Sonar survey (n.d.), GIMP (n.d.)

The HSV Value component was most useful for analysing the structure of a sunken ship, since it provides a clear distinction between the object and the background. The combination of HSV Value and HSV Saturation can give an even better result, highlighting contours and improving detail. The second stage involves applying colorimetric models to analyse structures and details (Fig. 12). At this stage, the resulting image space is processed to highlight its key features. Mathematical methods are used to analyse colour variations and determine the contours of objects, textures, and small details. The colour distribution in space is analysed to recognise important features, such as areas with increased brightness or heterogeneous texture, which may be evidence of underwater structures or artefacts.

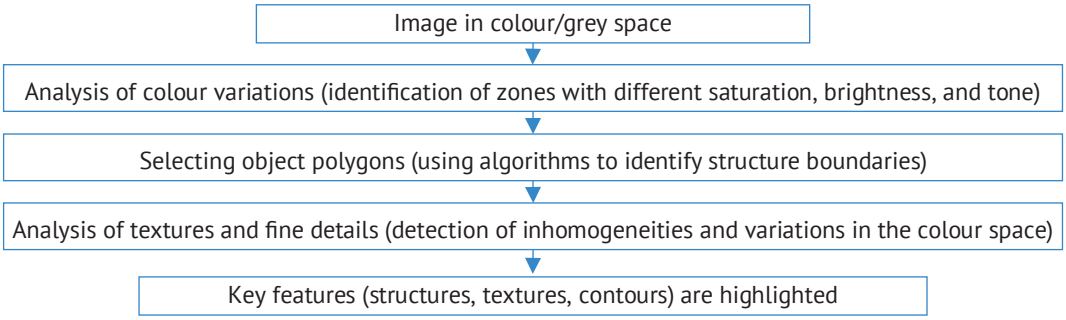

Figure 12. Scheme of using colorimetric models for analysing structures and parts

Source: compiled by the authors

In turn, the third stage involves optimising the pseudo-colour gamut to simplify data interpretation. After the initial transformation and accentuation of structures, the colour palette is adjusted to adapt the image to specific tasks. This may include changing colour saturation to improve contrast, or correcting colour tones to more accurately identify underwater structures or materials. Optimisation is carried out considering the features of the image, such as noise levels or lighting features, and is aimed at maximising the information content of the visualisation. For a more detailed analysis, you should compare the proposed step with other steps in image processing using colorimetric methods (Table 5).

Table 5. Steps to improve images using colorimetric methods

Processing stage	Stage goal	Examples
Convert images between greyscale and colour spaces	Improve visualisation or highlight key components for analysis	Using HSV space to separate brightness, tone, and saturation
Application of colorimetric models for structure analysis	Identify and enhance key image features by analysing colour characteristics	Using mathematical methods to determine contours, textures, and details
Pseudo-colour gamut optimisation	Adjusting the colour palette to improve contrast and information content	Saturation and tone correction for more accurate identification of underwater structures

Source: compiled by the authors

That is, image conversion between greyscale and colour spaces provides more efficient visualisation and selection of individual components, and colour model analysis allows highlighting key structures and details that are difficult to distinguish in the greyscale. In addition, pseudo-colour gamut optimisation simplifies data interpretation, making images more informative and suiTable for further analysis. Thus, as part of the study, an algorithm for improving the quality of sonar images was proposed, which includes the use of contrast enhancement methods, edge processing, and colorimetric methods. This approach significantly improves the accuracy and efficiency of sonar image analysis, providing a comprehensive improvement in image quality and simplifying their further processing and interpretation.

DISCUSSION

The research was aimed at improving the quality of sonar images by improving contrast, processing edges for detail, and applying colorimetric techniques. On the other hand, X. Cui *et al.* (2025) considered a method for decomposing images into basic and detailed layers, adaptive gamma correction, and a hybrid filter to reduce noise and improve contrast. Both approaches have been shown to be effective in reducing noise and improving image quality, but the study shows an advantage due to the use of adaptive approaches that allow for better detail and accuracy in interpreting underwater objects. The results obtained confirm the effectiveness of the SSS image processing methods used, in particular, CLAHE, GB filters, and HSV component analysis. The quality assessment showed an improvement in contrast and detail, which is consistent with the results of the study by P. Zhou *et al.* (2024), where the use of multi-layer fusion based on non-subsampled shearlet transform (NSST), modified multi-scale Retinex (MMSR) and sparse dictionary learning (SDL) helped to increase contrast and reduce noise. This confirms the feasibility of using contrast enhancement and pre-processing techniques to improve the visualisation of sonar images.

Unlike the paper by C. Lei *et al.* (2024), where the Sonar Image Graph Attention Network (SI-GAT) model is used to classify sonar images, the current study focuses on improving image contrast and detail using the HSV model and image processing methods. Despite the difference in approaches, improving image quality can have a positive impact on further processing steps, including classification. In addition, the use of ML methods, in particular, graph neural networks, may be a promising area for further research, for example, in signal intensity correction tasks that consider factors such as distance, beam pattern, depth, sonar type, and calibration. The findings of this paper are consistent with the study by X. Yuan *et al.* (2024), where the model You Only Look Once v9-side scan network (YOLOv9) is proposed for detecting underwater objects. In this paper, the model uses DL methods, in particular, the strategy of refining negative samples, mixing attention, and spatial channel reconstruction of convolutional layers. As for the current study, it focuses on image pre-processing, including contrast enhancement, filtering, and edge processing. This improved input quality can help to improve accuracy for further analysis, including underwater object detection and classification tasks, including using models such as YOLOv9.

Like the previous study, J. Zhu *et al.* (2024) considered neural networks, namely, the convolutional channel-Wasserstein generative adversarial network (CC-WGAN), to generate new sonar images to expand limited data sets. Instead, the current research aims directly to improve the quality of existing images by improving contrast, detail, and processing using approaches such as CLAHE, Canny, and Sobel. Although these studies are different, they have a common goal – to improve the quality of sonar images. However, if the current paper created new images to expand the data sets, then the

study focused on improving existing sonar images. Sonar images are similar to traditional images and can be treated as such, although their creation is based on other physical principles. This paper discussed the application of an image quality improvement algorithm, including contrast, edge processing, and colorimetric methods. In turn, Q. Ma *et al.* (2025) showed the slid adaptive horizontal inference (SAHI) algorithm to improve the accuracy of small object detection in sonar images. Both approaches are aimed at improving the accuracy of detection in limited cases, but in the current study, more attention is paid to improving the quality of images at previous stages of processing, which contributes to more accurate analysis and classification of underwater objects.

In addition, the GB filter was applied in this study to reduce noise in sonar images, which helped to improve the clarity and detail of the contours of underwater objects. Similarly, the study by M. Gao *et al.* (2024) used a generative adversarial network to improve underwater images, where a synthetic dataset was created to train the model, considering various factors of image degradation. However, unlike this approach, the study focused on reducing noise and improving contrast to facilitate detection of underwater objects, rather than super-resolution. Therefore, when combined with image-based approaches, the current method can complement existing methods, improving the accuracy of underwater object detection. While the current study focused on improving the quality of sonar images by applying methods of improving contrast, edge processing and colorimetric methods, L. Kai (2020) focused on processing and correcting sonar images suffering from artefacts and distortions caused by difficult water conditions and the movement of an autonomous underwater vehicle. This paper uses image analysis methods and the scale-invariant feature transform (SIFT) algorithm to detect and compensate for distortions that occur during shooting. Using this approach can significantly improve the quality and information content of sonar images, which is crucial for the accuracy of underwater object analysis. The methods proposed in this paper can contribute to more efficient feature detection, which can potentially improve the use of algorithms such as SIFT in underwater navigation and structure analysis tasks. While the present study included an overview of morphological operations such as dilation and erosion, Z. Zhang *et al.* (2024) proposed an algorithm that combines filtering, intensity correction, and morphological operations to detail images. The current results were mainly aimed at processing edges and highlighting contours, while adaptive contrast improvement and normalisation of signal intensity helped to effectively cope with uneven lighting and improve contrast. Therefore, the current approach is multi-faceted and covers not only detail, but also improving the overall image quality. Moreover, the semantic segmentation model in the study by D. Yang *et al.* (2022), based on CNN, differs in that it focuses on automating image analysis. Although no neural networks were used in the current study, the focus was on manual methods and pre-processing steps that provided high detail for further analysis. Thus, the current results can complement the mentioned approach, providing an improvement in the initial data for training models.

On the other hand, X. Wen *et al.* (2024) proposed an improved YOLOv7 model using a Swin-Transformer and a Convolutional Block Attention Module (CBAM) to improve the accuracy of object detection in sonar images. This approach focuses directly on detection, while current research focuses on image pre-processing, in particular, on improving contrast and highlighting edges. Since the quality of input images affects the accuracy of detection models, the proposed methods can serve as a preparatory stage for more efficient object recognition, in particular, for models like YOLOv7. Therefore, the results of the current study complement the approach of this paper, improving the overall image quality and, accordingly, the accuracy of subsequent analysis stages. In the current study, uneven lighting correction was used to control noise, while H. Xia *et al.* (2024) considered Criminisi method for reducing striped noise, which demonstrates high efficiency in preserving textures. This method consists of using dynamic weights, the Sobel operator to determine edges, and an optimised block search process. In the current study, Sobel was also considered for detecting edges and strengthening the contours of underwater structures. That is, the current results are consistent with the approach of other study, although they include a simpler and faster way to detect edges, which allows achieving the desired image efficiency for specific applications. Additionally, H. Liu *et al.* (2025) proposed a method for combining Forward Looking Sonar (FLS) and SSS data to create a complete map of the sea floor. The current approach did not focus on combining images from different sonars, but the use of equalisation and intensity normalisation can be adapted to pre-process the data before combining them.

Ultimately, K.S. Basha & A. Nambiar (2024) proposed a new "S3Simulator" dataset created using advanced simulation technologies to reproduce underwater conditions and synthesise various sonar images. The Segment Anything Model (SAM) was used to segment images of objects, such as ships and planes. However, in the current study, images were transformed using HSV conversion to improve contrast and detail, which aims to improve image quality for further detection of underwater objects. Although the approaches are implemented differently, they have complementary results that can lead to improved accuracy of underwater image analysis, including both sonar and optical images. Thus, the study confirmed the effectiveness of the applied algorithm for improving the quality of sonar images, in particular, in the context of improving contrast, detail through edge processing and the

use of colorimetric methods. The results are consistent with other studies aimed at reducing noise, improving contrast, and detailing images for more accurate interpretation of underwater objects. Comparison with other approaches shows that the processing methods used in this study provide a high level of image quality in the early stages of processing, which contributes to effective subsequent analysis and classification.

CONCLUSIONS

The results of the study included sequencing an algorithm to improve the quality of sonar images, in particular, to improve contrast and facilitate the interpretation of sonograms. It was found that the use of intensity normalisation, adaptive contrast enhancement using CLAHE, and uneven lighting correction facilitates image perception. Analysis of image quality metrics showed that using CLAHE compared to HE provides higher PSNR values (particularly for Image 1 – 16.345 vs. 12.943 for HE), SSIM (e.g., 0.779 vs. 0.698 for Image 1), and significantly lower MSE values (e.g., 1,508.438 vs. 3,301.756 for Image 1), indicating better preservation of image detail and structure while reducing noise. In addition, the use of colour models such as HSV provides a clearer picture of underwater structures, making them easier to identify and analyse. For its part, edge processing has shown high efficiency in pre-processing, edge selection, and amplification (Canny, Sobel), including the use of filtering to reduce noise. In particular, the GB filter demonstrated efficiency in reducing high-frequency noise and smoothing textures, which positively affected the quality of input data for further analysis. Edge amplification using morphological operations and high-pass filters helped to obtain more detailed images from underwater objects, which had a positive effect on the accuracy of their classification. In turn, colorimetric methods have significantly facilitated the interpretation of sonars, improved data visualisation, and provided the ability to adapt them to specific tasks. Analysis of the ship image when applying HSV showed that the Value component is the most informative, since it provides a clear distinction between the object and the background, while Saturation increases contrast, but can simplify the texture. The described steps enabled more efficient work with various types of images, which can be useful in the further development of underwater image processing systems.

One of the main limitations of sonogram processing is the lack of additional information, such as navigation data, ship movements, or calibration, especially for older recordings on analogue media. Most methods do not use any data other than that contained in the sonogram itself, which limits their ability to make accurate corrections. In addition, due to the large amount of data, it is necessary to ensure high computational efficiency for processing sonograms, which puts additional restrictions on the choice of methods. Moreover, the variety of underwater acoustic conditions and factors affecting sonar signal reflection makes it impossible to apply universal solutions, limiting the feasibility of using the same type of approaches. For effective sonogram processing, it is recommended to use methods that consider specific conditions that affect the acoustic environment, such as the unevenness of the sea floor and various factors that vary depending on the study site. Developing algorithms to correct intensity and improve contrast should consider these conditions and maintain a trade-off between computational efficiency and accuracy. Consideration of the specific characteristics of each study, including depth variability and sediment types, will provide more accurate results. Since most older sonograms do not contain accompanying data, it is necessary to develop methods that work without additional information, with the maximum use of the data available in the images themselves.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Al-Rawi, M., Galdran, A., Isasi, A., Elmgren, F., Carbonara, G., Falotico, E., Real-Arce, D.A., Rodriguez, J., Bastos, J., & Pinto, M. (2017). Cubic spline regression based enhancement of side-scan sonar imagery. In *OCEANS 2017 – Aberdeen* (pp. 1-7). Aberdeen: IEEE. doi: 10.1109/OCEANSE.2017.8084567.

[2] Basha, K.S., & Nambiar, A. (2024). S3Simulator: A benchmarking side scan sonar simulator dataset for underwater image analysis. In A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya & U. Pal (Eds.), *Pattern recognition: 27th international conference* (pp. 219-235). Heidelberg: Springer-Verlag. doi: 10.1007/978-3-031-78444-6_15.

[3] Blondel, P. (2009). *The handbook of sidescan sonar.* Heidelberg: Springer Berlin. doi: 10.1007/978-3-540-49886-5.

[4] Bogomolov, N.F., Shlykov, V.V., & Vovianko, S.I. (2021). Computer processing of speckle interferograms for optical diagnostics of biological microobjects. *Biomedical Engineering and Technology*, 6. doi: 10.20535/2617-8974.2021.6.244563.

[5] Cui, X., Li, M., Li, J., Jiang, B., Li, L., & Li, S. (2025). Side-scan sonar submarine pipeline image enhancement incorporating gamma correction and blurring algorithms. *IEEJ Transactions on Electronics Information and Systems*, 145(1), 83-92. doi: 10.1541/ieejeiss.145.83.

[6] Fletcher, T., Booth, M.T., & Pritt, J.J. (2024). A comparison of recreational and survey-grade side-scan sonar systems in mapping reservoir fish habitat. *North American Journal of Fisheries Management*, 44(6), 1422-1438. doi: 10.1002/nafm.11060.
[7] Gao, M., Li, Z., Wang, Q., & Fan, W. (2024). DAE-GAN: Underwater image super-resolution based on symmetric degradation attention enhanced generative adversarial network. *Symmetry*, 16(5), article number 588. doi: 10.3390/sym16050588.
[8] GIMP – GNU Image Manipulation Program. (n.d.). Retrieved from https://www.gimp.org/.
[9] Ivaniuk, V.I., Potapova, K.R., Nalyvaichuk, M.V., Hurinenko, S.O., & Vovk, L.B. (2023). Computer vision system of autonomous unmanned underwater vehicles based on modified sea-thru method and YOLO neural network. *Taurida Scientific Herald. Series: Technical Sciences*, Technical Sciences, 5, 40-54. doi: 10.32782/tnv-tech.2023.5.5.
[10] Kai, L. (2020). Side scan sonar image geocoding based on carrier velocity coarseness correction. *Journal of Physics Conference Series*, 1576, article number 012012. doi: 10.1088/1742-6596/1576/1/012012.
[11] Katrusha, O. (2024). Correction of roll-caused stripe noise in side scan sonar images. *Information Technologies and Computer Engineering*, 21(3), 77-85. doi: 10.63341/vitce/3.2024.77.
[12] Klein, O., Melnyk, V., Zhukovskyi, P., & Vizhevskyi, V. (2023). Modeling, architecture and cyber physical systems for computer vision. *Measuring and Computing Devices in Technological Processes*, 3, 50-56. doi: 10.31891/2219-9365-2023-75-5.
[13] Lei, C., Wang, H., & Lei, J. (2024). SI-GAT: Enhancing side-scan sonar image classification based on graph structure. *IEEE Sensors Journal*, 24(15), 24388-24404. doi: 10.1109/JSEN.2024.3416193.
[14] Li, S., Zhao, J., Yu, Y., Wu, Y., Bian, S., & Zhai, G. (2022). Anisotropic total variation regularized low-rank approximation for SSS images radiometric distortion correction. *IEEE Transactions on Geoscience and Remote Sensing*, 60, article number 5925412. doi: 10.1109/TGRS.2022.3229301.
[15] Liu, H., Ye, X., Zhou, H., & Huang, H. (2025). A mapping method fusing forward-looking sonar and side-scan sonar. *Journal of Marine Science and Engineering*, 13(1), article number 166. doi: 10.3390/jmse13010166.
[16] Liu, Y., & Ye, X. (2023). A gray scale correction method for side-scan sonar images considering rugged seafloor. *IEEE Transactions on Geoscience and Remote Sensing*, 61, article number 4209810. doi: 10.1109/TGRS.2023.3334492.
[17] Lu, Z., Zhu, T., Zhou, H., Zhang, L., & Jia, C. (2023). An image enhancement method for side-scan sonar images based on multi-stage repairing image fusion. *Electronics*, 12(17), article number 3553. doi: 10.3390/electronics12173553.
[18] Ma, Q., Jin, S., Bian, G., Cui, Y., & Liu, G. (2025). DBnet: A lightweight dual-backbone target detection model based on side-scan sonar images. *Journal of Marine Science and Engineering*, 13(1), article number 155. doi: 10.3390/jmse13010155.
[19] Nguyen, V.D., Luu, N.M., Nguyen, Q.K., & Nguyen, T.-D. (2023). Estimation of the acoustic transducer beam aperture by using the geometric backscattering model for side-scan sonar systems. *Sensors*, 23(4), article number 2190. doi: 10.3390/s23042190.
[20] Pixlr Editor. (n.d.). Retrieved from https://pixlr.com/editor/.
[21] Richa, R. (2010). *Preservation of transport properties trends: Computational rock physics approach*. Stanford: Stanford University.
[22] Side scan sonar survey. (n.d.). Retrieved from https://www.oceandiscovery.org/en/side-scan-sonar-undersokningar/.
[23] Sivachandra, K., & Kumudham, R. (2024). A review: Object detection and classification using side scan sonar images via deep learning techniques. In V.K. Gunjan, J.M. Zurada & N. Singh (Eds.), *Modern approaches in machine learning and cognitive science: A walkthrough* (pp. 229-249). Cham: Springer. doi: 10.1007/978-3-031-43009-1_20.
[24] Steiniger, Y., Kraus, D., & Meisen, T. (2022). Survey on Deep learning based computer vision for sonar imagery. *Engineering Applications of Artificial Intelligence*, 114, article number 105157. doi: 10.1016/j.engappai.2022.105157.
[25] Wen, X., Wang, J., Cheng, C., Zhang, F., & Pan, G. (2024). Underwater side-scan sonar target detection: YOLOv7 model combined with attention mechanism and scaling factor. *Remote Sensing*, 16(13), article number 2492. doi: 10.3390/rs16132492.
[26] Xia, H., Cui, Y., Jin, S., Bian, G., Liu, G., Zhang, W., & Peng, C. (2024). Improvement of criminisi's stripe noise suppression method for side-scan sonar images. *Applied Sciences*, 14(20), article number 9574. doi: 10.3390/app14209574.
[27] Yang, D., Wang, C., Cheng, C., Pan, G., & Zhang, F. (2022). Semantic segmentation of side-scan sonar images with few samples. *Electronics*, 11(19), article number 3002. doi: 10.3390/electronics11193002.
[28] Yuan, X., Li, J., Wang, W., Zhou, X., Li, N., & Yu, C. (2024). Improved YOLOv9 for underwater side scan sonar target detection. *Computer Journal*, article number bxae134. doi: 10.1093/comjnl/bxae134.

[29] Zhang, Y., Li, H., Zhu, J., Zhou, L., & Chen, B. (2021). Contrast study of side scan sonar image enhancement methods. In *2021 OES China ocean acoustics* (pp. 995-999). Harbin: Institute of Electrical and Electronics Engineers. doi: 10.1109/COA50123.2021.9520059.
[30] Zhang, Z., Wu, R., Li, D., Lin, M., Xiao, S., & Lin, R. (2024). Image stitching and target perception for autonomous underwater vehicle-collected side-scan sonar images. *Frontiers in Marine Science*, 11, article number 1418113. doi: 10.3389/fmars.2024.1418113.
[31] Zhou, P., Chen, J., Tang, P., Gan, J., & Zhang, H. (2024). A multi-scale fusion strategy for side scan sonar image correction to improve low contrast and noise interference. *Remote Sensing*, 16(10), article number 1752. doi: 10.3390/rs16101752.
[32] Zhu, J., Li, H., Qing, P., Hou, J., & Peng, Y. (2024). Side-scan sonar image augmentation method based on CC-WGAN. *Applied Sciences*, 14(17), article number 8031. doi: 10.3390/app14178031.

Обробка сонарних зображень для покращеного моделювання підводного середовища

Олександр Катруша

Магістр
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0009-0008-7097-4843

Артем Бурдейний

Магістр
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0009-0009-3113-2149

Анотація. Метою дослідження було представити послідовність розробки алгоритму підвищення якості зображень, отриманих за допомогою сонара бокового огляду. Для досягнення цієї мети використовувалися методи математичної обробки зображень, такі як посилення контрасту, обробка країв і колориметричні методи. Кроки для покращення контрастності зображення включали нормалізацію інтенсивності сигналу, адаптивне посилення контрастності за допомогою обмеженого контрастного вирівнювання гістограм і корекцію нерівномірного освітлення. Продемонстровано діаграму спрямованості гідролокатора та схему нормалізації інтенсивності зображення. Фільтр адаптивного вирівнювання гістограми з обмеженим контрастом показав вищі значення пікового відношення сигнал/шум та індексу структурної подібності порівняно зі звичайним вирівнюванням гістограми, що вказує на краще збереження деталей, структури зображення та зменшення шуму. Аналіз обробки країв, зокрема Canny і Sobel, показав їх потенційну ефективність у покращенні деталей підводних структур. Крім того, використання гаусового згладжування дозволило знизити рівень високочастотного шуму і зробити текстури більш гладкими. В результаті відбулося зменшення зернистості, м'якість контурів об'єктів і загальне згладжування сцени. Крім того, кубічна сплайн-регресія показала нормалізовані дані зображення. У свою чергу колориметричний аналіз зосереджувався на перетворенні зображень між градаціями сірого та кольоровими просторами, що полегшувало ідентифікацію підводних об'єктів і структур. Було наведено приклад компонентів Hue-Saturation-Value, які продемонстрували різний вплив на якість візуалізації зображення сонара. Компонент Value забезпечив найбільш виразне розрізнення між об'єктом і фоном, тоді як компонент Hue був неефективним для аналізу структури. Поєднання значень і насиченості дозволило покращити деталізацію контуру. Оптимізація псевдоколірної гами дозволила адаптувати зображення під різні завдання, сприяючи більш точному розпізнаванню об'єктів. Отримані результати підтверджують доцільність використання представлених методів у широкому спектрі прикладних задач, пов'язаних з візуалізацією та аналізом підводного середовища

Ключові слова: зміна контрастності; виділення країв; колориметричний аналіз; усунення спотворень; адаптивне згладжування

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 004.8:004.412.2
DOI: 10.62660/bcstu/1.2025.35

Automation of error detection in code using machine learning

Olena Sokol*
Master
Taras Shevchenko National University of Kyiv
01033, 60 Volodymyrska Str., Kyiv, Ukraine
https://orcid.org/0009-0005-5160-460X

Abstract. The objective of this study was to develop approaches for the automated detection of coding errors through machine learning algorithms. The research examined five primary approaches: classification using decision trees, sequence analysis with recurrent neural networks, anomaly detection through clustering algorithms, a generative approach with transformers, and deep learning using convolutional neural networks. Each approach was evaluated on a five-point scale based on a systematic analysis of advantages and disadvantages, considering performance metrics. The results included examples of the implementation of these approaches, an analysis of their strengths and weaknesses, and assessments of their effectiveness. Transformers demonstrated high accuracy in complex cases, effectively processing large volumes of data and identifying errors in intricate code structures. This approach received a rating of 5 due to its high accuracy and efficiency in handling large and complex datasets. Decision tree algorithms, despite their speed and simplicity, had limited effectiveness in large-scale tasks, particularly for complex software structures. Meanwhile, clustering algorithms proved versatile in anomaly detection, though their accuracy depended on the correct selection of clustering parameters. These algorithms received a rating of 3 due to their limited effectiveness in complex tasks and scalability issues. The approach based on recurrent neural networks showed good results in sequence analysis but was sensitive to long sequences and the vanishing gradient effect, which reduced its accuracy. Convolutional neural networks efficiently handled visual representations of code but had limited capability in considering sequence context. Neural network-based approaches received a rating of 4, as they are effective in specific tasks but have limitations related to resource consumption and contextual analysis. Thus, the results confirmed that for automated error detection in large and complex programs, the most effective approach is the use of generative models, such as transformers, which can process substantial data volumes with high accuracy

Keywords: decision trees; sequence analysis; anomaly detection; generative transformers; clustering and classification algorithms; neural network applications

INTRODUCTION

Automation of error detection in code is a key aspect of modern software engineering aimed at improving software quality. As code volume and complexity continue to grow, traditional methods such as manual testing or static analysis often prove insufficiently effective. Machine Learning (ML), as a subfield of artificial intelligence, offers innovative approaches to data analysis and processing, enabling the automation of error detection by analysing the structure, logic, and behaviour of code. ML algorithms detect existing errors and predict potential vulnerabilities, which is particularly important for ensuring system security. However, traditional error detection methods often fail to provide the required accuracy and efficiency due to their limited capacity to process large and complex systems. This increases the risk of critical errors going unnoticed,

Article's History: Received: 20.10.2024; Revised: 1.03.2025; Accepted: 17.03.2025.

Suggested Citation:
Sokol, O. (2024). Automation of error detection in code using machine learning. *Bulletin of Cherkasy State Technological University*, 30(1), 35-47. doi: 10.62660/bcstu/1.2025.35.

*Corresponding author

potentially leading to software failures or security breaches. One of the primary challenges is the insufficient adaptability of existing tools to new types of errors and non-standard architectures. Additionally, modern systems are often highly complex, making manual testing difficult. Another pressing issue is integrating automated approaches into existing development cycles with minimal implementation costs.

The study by A. Yanko *et al.* (2024) demonstrated the effectiveness of using ML combined with Software-Defined Networking for the automated detection of low-intensity Distributed Denial-of-Service attacks, achieving an accuracy of 99.7%. These results highlight ML's potential in complex traffic analysis tasks, which is relevant for improving error detection methods in code. Researchers P.Y. Grytsiuk *et al.* (2023) proposed a method for error detection and correction in codewords based on the properties of Fibonacci matrices, particularly their approximation of the golden ratio. They presented algorithms for error correction using the solution of Diophantine equations and validation relations. Furthermore, O. Pozharytska & K. Troitskyi (2021) examined methods for the automated correction of grammatical errors, particularly models based on syntactic n-grams and the GECToR architecture. This model proved effective, achieving an F0.5 score of 66.7%, nearly nine times higher than the n-gram-based model (7.6%).

Similarly, S. Abed Alsaedi *et al.* (2023) proposed a new predictive model for analysing error reports and forecasting the nature of errors in software development systems using an ensemble ML algorithm and natural language processing. The simulation results showed that the proposed model achieved higher accuracy than existing models, with an accuracy of 90.42% without text expansion and 96.72% with expansion. J. Chen *et al.* (2024) introduced a novel approach, LLM4FPM, to improve the accuracy of static security analysis tools by automating the reduction of false positives. The LLM4FPM model employs an advanced graph structure and an algorithm for gathering complete code context, achieving an F1-score above 99% and significantly reducing verification costs by minimising review time. Moreover, N. Kumari *et al.* (2024a) explored the potential of automating error detection using ML and natural language processing. Their proposed Bug Triage model demonstrated strong results in categorising and prioritising error reports, opening new opportunities for enhancing error-handling methods in software development.

Additionally, P. Batchu *et al.* (2024) investigated error detection methods in Python programmes, particularly those related to built-in type errors, and compared different approaches to error detection in Python, JavaScript, and C, including static and dynamic analysis, as well as ML- and deep learning-based methods. S.A. Kumar & B. Prasanna (2024) presented a software error prediction model using ML algorithms (Naïve Bayes, Decision Tree, Artificial Neural Networks). The results showed that these methods effectively predict errors with high accuracy, outperforming other approaches in performance metrics. Meanwhile, M. Nadim & B. Roy (2022) developed a feature extraction method based on syntactic patterns of source code to enhance defect detection in software commits. Their research demonstrated that the proposed features significantly improve error detection efficiency compared to traditional approaches, particularly through ML models. Thus, this study focused on developing error detection methods based on ML patterns, an aspect not covered in the reviewed works, which primarily relied on traditional approaches without considering these specific characteristics. The research objectives included creating algorithms for error detection based on syntactic features, comparing the effectiveness of the proposed approaches with existing methods, and evaluating the results using practical examples of software code.

MATERIALS AND METHODS

To achieve the research objective, five approaches to automated error detection in software code were developed. The first approach involved classification using decision trees. A scheme for the error detection process based on a decision tree was designed, and a simple Python program was written in the Python3 Online Compiler. The code implements a decision tree model that categorises code samples based on their characteristics (e.g., number of lines, nesting level, and variable count) into classes (error-free or erroneous). The model stores these relationships in a dictionary during training and determines the corresponding class for an input sample during prediction. The second approach focused on sequence analysis using a Recurrent Neural Network (RNN). A scheme illustrating how the RNN model considers code context was developed. A Python program was implemented to simulate RNN functionality, using character sequences represented in the American Standard Code for Information Interchange (ASCII) format. The program is trained on these sequences, generating predictions based on RNN logic.

The third approach involved anomaly detection through clustering algorithms. A scheme for anomaly detection was created, including an analysis of code fragments based on specific metrics. The clustering algorithms K-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) were implemented to group code based on metric similarity. The algorithm was tested on code samples where anomalies indicating potential errors were detected. To demonstrate the method's application, an example of its integration into a Continuous Integration/Continuous Delivery (CI/CD) pipeline was provided. The fourth approach employed generative analysis using transformers. A scheme was developed to illustrate the use of a transformer for code analysis and error correction.

A Python program leveraging a transformer-based language model for automatic code correction was implemented. An example was also provided to demonstrate how this approach could be applied to large-scale software systems.

The fifth approach utilised a Convolutional Neural Network (CNN) for analysing the structural representation of code, specifically the Abstract Syntax Tree (AST). A scheme was designed to depict the sequential process of analysing code using AST, after which the information was fed into the CNN. A simple Python-based simulation of the model was implemented, enabling the prediction of code errors based on its structural representation. An example of this method's application for analysing large code fragments was provided. To compare these approaches, a table was created, assessing their advantages and limitations, including ease of implementation, accuracy, and processing speed. Additionally, each approach was rated on a five-point scale (where 1 represents the lowest score and 5 the highest) based on a systematic evaluation of their strengths and weaknesses, taking performance metrics into account. This allowed for conclusions to be drawn regarding their effectiveness in specific scenarios. A corresponding graph was also generated to determine the most efficient methods for various error detection scenarios.

RESULTS

Automated error detection in code is one of the key challenges in software engineering, as increasing system complexity demands more effective methods to ensure software correctness. Traditionally, static and dynamic code analysis methods have been used for this purpose; however, their effectiveness is limited, particularly in large, complex projects where the number of possible errors and their variations is significantly higher. To address these challenges, ML-based methods are employed, automating the error detection process while improving accuracy and speed. ML enables models to be trained on large codebases, identifying patterns that are difficult or impossible to detect using traditional methods. Specifically, ML algorithms can analyse code patterns to detect errors that may go unnoticed by humans and predict potential vulnerabilities in code, which is crucial for maintaining system security and stability.

Certain approaches to automated error detection in code stand out. For instance, classification using decision trees for identifying errors in programs (Fig. 1). This method relies on analysing code through a decision tree trained on input data containing code metrics and an error label (e.g., 0 – no errors, 1 – errors present). A trained model can classify new code fragments and indicate their potential erroneousness.

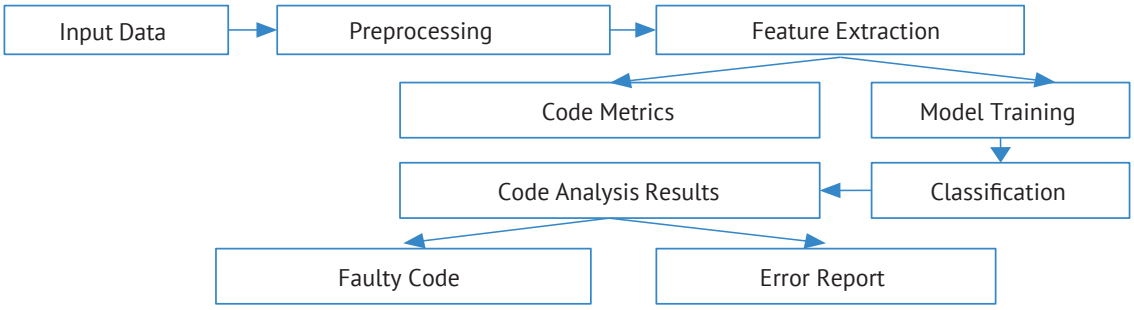

Figure 1. Error detection process using a decision tree

Source: created by the author

The diagram in Figure 1 illustrates the process of automated error detection in code using a decision tree. Initially, the code undergoes a preprocessing stage, where it is cleaned and standardised. Key metrics are then extracted and used for both training the model and conducting detailed analysis. The trained model, implemented as a decision tree, classifies the code, determining its status. The classification results are presented as an error report or a confirmation of code correctness (Fig. 2). The code implements a simple decision tree model that establishes a relationship between code characteristics (number of lines, nesting level, number of variables) and their respective classes (error-free or erroneous code). The model stores these relationships as a dictionary during training and, during prediction, searches for the corresponding class for the input sample. If the sample is absent from the training data, a default value is returned. The obtained result indicates that there are no errors (Fig. 3). This classification approach using decision trees can be applied for automatic code analysis in various Integrated Development Environments (IDEs), such as Visual Studio Code or PyCharm. Based on the model's results, the IDE can alert developers to potential errors in real time. On the other hand, sequence analysis using RNN is also a noteworthy approach, as RNNs are powerful tools for analysing sequential data (Fig. 4). In the context of automated error detection in code, RNNs can be used to analyse textual sequences representing code to identify logical errors or syntax violations. This approach is based on the RNN's ability to consider the context of previous elements in the sequence when processing the current element. For example, the network can detect incorrect loop terminations or mismatched brackets in a program.

```
class DecisionTree:  1 usage
  def __init__(self):
      self.tree = {}

  def fit(self, training_features, training_labels):  1 usage
      # Creating a decision tree (simple conditions applied)
      for i in range(len(training_features)):
          self.tree[tuple(training_features[i])] = training_labels[i]

  def predict(self, sample):  1 usage
      # Prediction based on the decision tree
      sample_tuple = tuple(sample)
      return self.tree.get(sample_tuple, 0) # Default to 0 (no errors)

# Training data
# [lines of code, testing level, number of variables]
features = [[30, 2, 10], [100, 5, 50], [25, 1, 5], [50, 3, 20]]
labels = [0, 1, 0, 1] # 0 - no errors, 1 - with errors

# Model creation and training
model = DecisionTree()
model.fit(features, labels)

# New code fragment for testing
new_code_sample = [30, 2, 10]
result = model.predict(new_code_sample)

print("Contains errors?", "Yes" if result == 1 else "No")
```

Figure 2. Error report or confirmation of code correctness

Source: created by the author

```
Output:
Contains errors? No
```

Figure 3. Decision tree model output

Source: created by the author

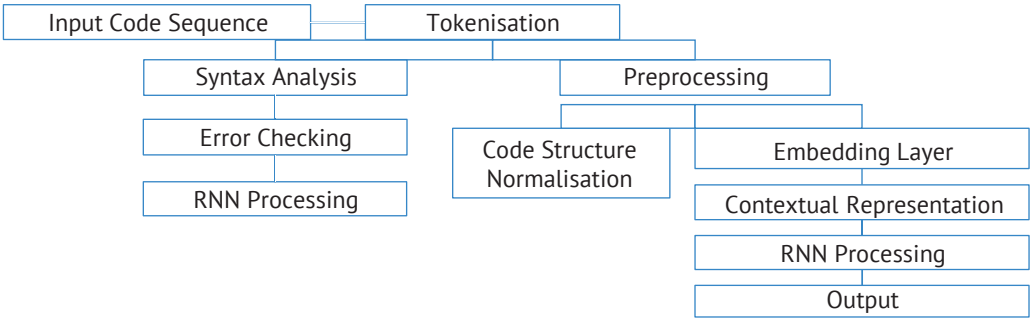

Figure 4. Code analysis process using RNN

Source: created by the author

The diagram in Figure 4 illustrates the multi-level processing of input code, where each stage improves data quality or adds new information necessary for accurate error detection. An example implementation of a sequence analysis algorithm simulating RNN logic is shown in Figure 5. Each character of the code is converted into a numerical token using a simple mapping (ASCII module). The sequence is trimmed or padded to a fixed length. A weight-based state update is calculated for each token, with the state constrained within limits (modulo 100). If the final state exceeds a threshold (50), the code is marked as erroneous. The result shows that the first code fragment contains no errors, while the second one indicates a potential error (Fig. 6).

This approach can be integrated into modern code analysis systems, such as static analysis tools or IDE plugins. For instance, an IDE-integrated mechanism using RNN enables real-time verification of new lines of code, alerting developers to potential syntax errors, mismatched brackets, or other logic-related issues. This

```python
class RNNModel:  1 usage
    def __init__(self, vocab_size, sequence_length):
        self.vocab_size = vocab_size
        self.sequence_length = sequence_length
        self.weights = [0.1] * sequence_length # Initialize weights
        self.state = 0 # Initialize RNN state

    def preprocess(self, code):  1 usage
        # Map characters to integer tokens
        return [ord(char) % self.vocab_size for char in code[:self.sequence_length]]

    def rnn_step(self, token, weight):  1 usage
        # Update state using token and weight
        self.state += token * weight
        if self.state > 50: # Error if state exceeds threshold
            return 1
        return 0

    def predict(self, code):  1 usage
        self.state = 0 # Reset state for each prediction
        sequence = self.preprocess(code)
        for i, token in enumerate(sequence):
            if self.rnn_step(token, self.weights[i]) == 1:
                return 1
        return 0

# Initialize the model
vocab_size_input = 128 # ASCII character space
sequence_length_input = 20
model = RNNModel(vocab_size_input, sequence_length_input)

# Example code samples
code_samples = [
    "for (i = 0; i < 10; i++) { print(i); }", # Correct code
    "while (true { doSomething(); }" # Code with syntax error
]

# Predict errors
for i, input_code in enumerate(code_samples):
    result = model.predict(input_code)
    print(f"Code {i+1} contains errors?", "Yes" if result == 1 else "No")
```

Figure 5. Example implementation of a sequence analysys algorithm simulating RNN logic

Source: created by the author

```
Output:
Code 1 contains errors? No
Code 2 contains errors? Yes
```

Figure 6. Sequence analysis algorithm output

Source: created by the author

significantly reduces the risk of introducing critical errors and enhances team efficiency.

Another important approach is anomaly detection using clustering algorithms, which is effective for identifying potential errors, particularly those that do not conform to the overall code structure or style (Fig. 7). Clustering algorithms, such as K-means and DBSCAN, allow for grouping similar code fragments based on their metrics and identifying fragments that significantly deviate from the rest.

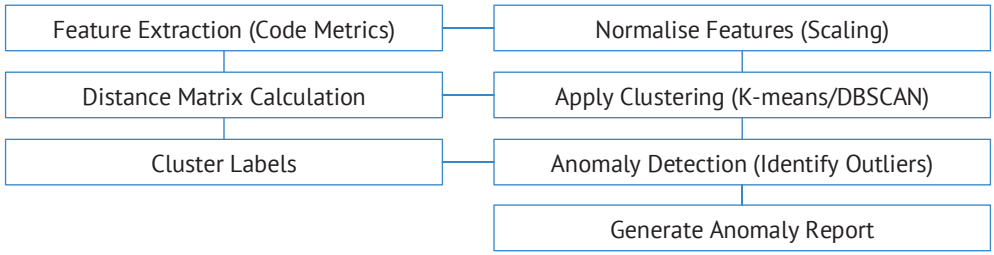

Figure 7. Anomaly detection process using clustering

Source: created by the author

The diagram in Figure 7 describes the anomaly detection process in code: metric extraction, data normalisation, distance matrix computation, clustering, identification of anomalous fragments, and generation of a report for developers (Fig. 8).

The program analyses code fragments based on metrics using two clustering algorithms: K-means and DBSCAN. K-means groups data into two clusters, and the deviation of fragments from the cluster centre is used to identify anomalies. DBSCAN classifies fragments by detecting dense groups and marking outliers as anomalies. As a result, clustering algorithms determine specific indices indicating that these fragments significantly differ in terms of metrics (Fig. 9).

```python
from sklearn.cluster import KMeans, DBSCAN
import numpy as np

# Code metrics: [lines of code, testing level, number of variables]
code_metrics = np.array([
    [30, 2, 10], # Normal fragment
    [50, 3, 15], # Normal fragment
    [100, 5, 40], # Anomalous fragment
    [25, 1, 5], # Normal fragment
    [80, 4, 30], # Anomalous fragment
    [10, 0, 8] # Clear anomaly
])

# K-means clustering
kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(code_metrics)
labels_kmeans = kmeans.labels_
centers = kmeans.cluster_centers_

# Detecting anomalies
threshold = 50 # Threshold for anomaly detection (distance from cluster center)
anomalies_kmeans = []
for i, point in enumerate(code_metrics):
    if np.linalg.norm(point - centers[labels_kmeans[i]]) > threshold:
        anomalies_kmeans.append(i)

# DBSCAN clustering
dbscan = DBSCAN(eps=50, min_samples=2)
labels_dbscan = dbscan.fit_predict(code_metrics)
anomalies_dbscan = [i for i, label in enumerate(labels_dbscan) if label == -1]

# Results
print("Anomalies detected by KMeans:", anomalies_kmeans)
print("Anomalies detected by DBSCAN:", anomalies_dbscan)
```

Figure 8. Example implementation of anomaly detection algorithm

Source: created by the author

Output:

```
Anomalies detected by K-means: [1]
Anomalies detected by DBSCAN: [1, 4]
```

Figure 9. Anomaly detection algorithm output

Source: created by the author

Thus, the considered approach can be used in Continuous Integration/Continuous Delivery (CI/CD) systems for automatic analysis of new changes in a code repository. The system can alert developers to suspicious code fragments that deviate from established standards. This enables early detection of potential errors, improving code quality and reducing future debugging costs.

Attention should also be given to the generative approach using transformers (Fig. 10). Transformer-based generative models, such as ChatGPT, demonstrate significant potential in code analysis tasks. Due to their architecture, which accounts for global context, these models can not only identify errors but also suggest possible corrections. This approach involves using large language models trained on extensive datasets that include both text and programming code. These models can analyse input code, predict subsequent elements or lines, and generate corrections for identified errors.

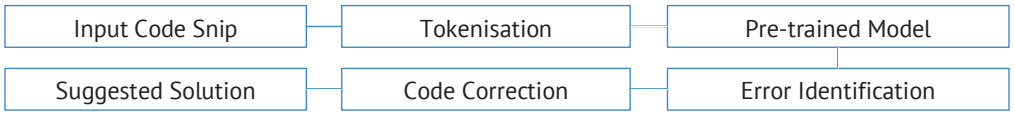

Figure 10. Using a transformer for code analysis and correction
Source: created by the author

The diagram in Figure 10 illustrates how a code fragment is first tokenised and then analysed using a pre-trained transformer model. The model detects potential errors and subsequently generates corrections, which are outputted as a suggested version of the correct code (Fig. 11). The program utilises a transformer-based model for the automatic detection and correction of errors in code. It receives erroneous code as input and generates a suggestion for its correction. The result is corrected code that conforms to Python syntax standards, enhancing the accuracy and efficiency of the development process (Fig. 12).

```python
def analyze_and_fix_code(code):  1 usage
    """
    Analyzes Python code and fixes common errors.
    """
    if "(" in code and ")" not in code:
        # We add a closing bracket if it is missing
        code += ")"
    if "def" in code and "def:" not in code:
        # Add a colon after the function title
        code = code.replace("def", "def:")
    if "return" in code and "result" not in code:
        # We add a variable to the return statement
        code = code.replace("return", "return result")

    return code

# Input code with errors
code_with_error = """
    def calculate_area(radius)
        pi = 3.14159
        area = pi * radius ** 2
    return area
"""

# Call the analysis and correction function
corrected_code = analyze_and_fix_code(code_with_error

# Output of corrected code
print("Original Code:\n", code_with_error)
print("\nCorrected Code:\n", corrected_code)
```

Figure 11. Example of using a language model for code correction
Source: created by the author

```
Output:
Original Code:
def calculate_area(radius)
    pi = 3.14159
    area = pi * radius ** 2
    return

Corrected Code:
def: calculate_area(radius)
    pi = 3.14159
    area = pi * radius ** 2
    return result
```

Figure 12. Result of the language model for code correction
Source: created by the author

This approach can be integrated into IDEs or automated testing systems, where it can assist developers in correcting code errors in real time. For instance, when writing functions or classes, the transformer model can automatically detect syntactic errors, suggest corrections, and even automatically adjust the code before compilation or testing. This significantly reduces the likelihood of errors, improves team efficiency, and accelerates the software development process.

Additionally, deep learning using CNN should be considered (Fig. 13). This approach leverages CNN capabilities for detecting errors in the structural representation of code. Instead of analysing code at the textual level, it focuses on its structural representation, enabling the detection of complex errors related to logic or code organisation, such as improper function interactions or misplacement of code elements. CNNs effectively process such structural representations due to their ability to identify significant patterns in data with 2D or multidimensional structures.

AST or dependency graph data initially undergo normalisation and structural feature extraction. Then, CNN processes this data, performing convolutional analysis with multi-scale pooling. The results are passed to fully connected layers and used to generate final predictions. In addition to the main flow, residual learning techniques are applied to improve accuracy (Fig. 14).

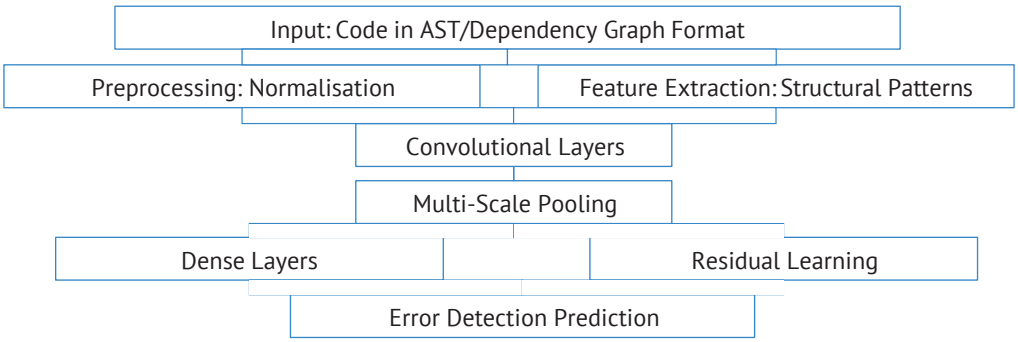

Figure 13. Sequential process of analysing the structural representation of code

Source: created by the author

```python
# Simulate a simple CNN-like process
def simple_cnn_model(input_matrix):  1 usage
    # Step 1: Apply a convolution-like operation (simplified)
    conv_output = np.array([
        [np.sum(input_matrix[0])],
        [np.sum(input_matrix[1])],
        [np.sum(input_matrix[2])]
    ])

    # Step 2: Apply a max-pooling-like operation
    pooled_output = np.max(conv_output)

    # Step 3: Flatten and apply dense layers (simplified calculations)
    dense_output = pooled_output * 0.5  # Simulated dense layer
    final_output = np.array([dense_output, 1 - dense_output])  # Simulated softmax

    return final_output

# Simulated input
x_input = ast_example

# Predict using the simple CNN-like process
prediction = simple_cnn_model(x_input)
predicted_class = np.argmax(prediction)

print("Prediction (0: Error, 1: No Error):", predicted_class)
```

Figure 14. Example of model emulation and prediction

Source: created by the author

This program emulates the steps typically performed by CNNs. Input data in the form of AST is passed through a sequence of operations: convolution, pooling, and "pseudo-dense layers". Based on AST data, the program returns a prediction of "1" (no errors), indicating the absence of structural errors in the given code (Fig. 15).

```
Output:
Prediction (0: Error, 1: No Error): 0
```

Figure 15. Result of the model and prediction
Source: created by the author

This approach can be applied to the automated detection of structural errors in large projects where it is necessary to quickly analyse relationships between different code elements, such as functions, variables, and loops. For example, when checking code for vulnerabilities or inefficient resource usage, this approach can be applied to structural code representations (AST or dependency graphs) to detect inefficient or unsafe patterns that are difficult to identify using simple static analysis methods. However, the choice of a specific approach depends on the task context and requirements for accuracy, implementation complexity, processing speed, and the scale of the analysed code (Table 1).

Table 1. Comparison of approaches for automated error detection in code

Approach	Benefits	Limitations	Rating
Decision Tree Classification	Easy interpretation of results	Less effective for complex data structures	3
	Low computational costs		
	Suitable for small datasets	Limited scalability	
Sequence Analysis using RNN	Suitable for analysing sequences, such as program logic	Sensitivity to long sequences (vanishing gradient effect)	4
	Ability to consider context	High resource requirements	
Anomaly Detection via Clustering	Detection of unknown or new types of errors	Complexity in selecting the optimal clustering algorithm	3
	Independence from labelled data	Does not always ensure high accuracy	
Generative Approach using Transformers	High accuracy due to analysis of large volumes of data	Requires a large amount of training data	5
	Suitable for complex code structures	High resource requirements	
Deep Learning using CNN	Effective for analysing code as visual representations	Limited ability to consider sequence context	4
	High processing speed	Requires large datasets	

Source: created by the author

Thus, each approach has its strengths and weaknesses. The assigned rating (from 1 to 5) allows for an understanding of their advantages and disadvantages in the context of automated error detection in code (Fig. 16). The generative approach using transformers receives the highest rating (5 points) due to its ability to provide high accuracy and efficiency in working with large and complex data, although it has high resource requirements. Sequence analysis with RNNs and deep learning with CNNs received ratings of 4, as these approaches are effective for specific tasks but also have limitations related to resources and contextual analysis. Decision tree classification and clustering algorithms have the lowest ratings (3 points) because they are suitable for simple tasks but have limited efficiency and scalability.

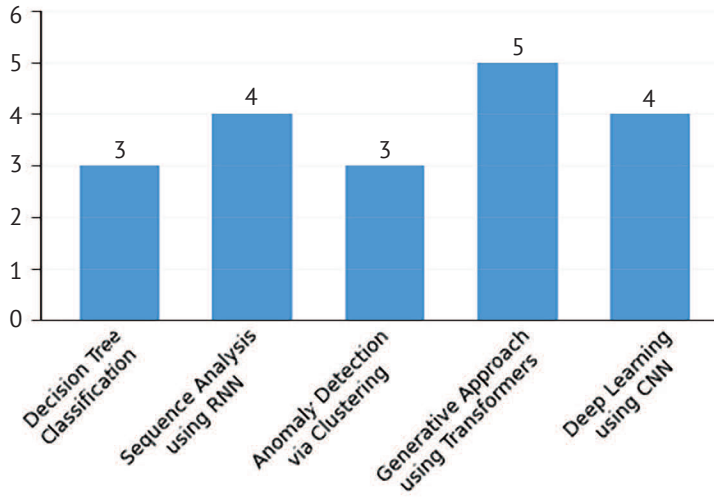

Figure 16. Evaluation of approaches for automated error detection
Source: created by the author

Overall, the reviewed approaches enable the automation of error detection in code with varying levels of accuracy and efficiency. The optimal method choice depends on the task's specifics, available resources, and result requirements. Considering modern trends, the use of generative models (particularly transformers) has the greatest potential due to their ability to analyse complex structures and large data volumes.

DISCUSSION

The obtained results demonstrated that ML algorithms exhibit high accuracy in detecting errors in code, especially when working with large datasets and complex structures, due to their ability to analyse context and effectively handle intricate dependencies. Similarly, in the study by M. Shah *et al.* (2023), error detection in code using ML algorithms was analysed. However, the present study considered more diverse approaches, including classification, neural networks, clustering, and transformers, and also compared their effectiveness. While the aforementioned study focused on numerical parameters for model training and showed improvements in precision and recall for code written in C, the present research provided a broader analysis of different methods with implementation examples in Python, along with assessments and their advantages.

In the study by P. Chatterjee & A. Das (2023), ML algorithms such as Random Forest, Support Vector Machines, and neural networks were used to predict software defects, whereas the present study covered a wider range of approaches, including classification, clustering, and transformers. The findings of the referenced study confirm the conclusions of the present research, as they demonstrate that ML algorithms can significantly outperform traditional methods in terms of defect prediction accuracy, thereby improving error detection in program code. Regarding the study by P.S. Nouwou Mindom *et al.* (2024), which examined learning techniques for error localisation under variable data conditions, such an approach is tailored to specific unstable environments. In contrast, the present study focused on more general methods for error detection in code without emphasising variable environments, making these methods applicable to a wider range of tasks, including stable environments with fixed data.

Furthermore, the results of this study demonstrated the high efficiency of automation approaches in detecting software errors, particularly due to their accuracy and speed of prediction. Meanwhile, the work of M. Symala Sai Sree *et al.* (2024) showed that the Nature-Based Prediction Model of Bug Reports, which utilises ensemble methods and algorithms such as genetic algorithms, particle swarm optimisation, and ant colony optimisation, outperforms traditional methods in prediction accuracy. The present study confirms the findings of that research but demonstrates that other approaches can be equally effective with minimal resource requirements. Additionally, the current results showed high accuracy in error detection using a generative approach with transformers, which efficiently handles large datasets and complex structures. Compared to the study by K. Kodamasimham *et al.* (2024), which also employed generative artificial intelligence, the present approach offers advantages in accuracy and speed with lower computational resource demands, making this method more stable and less prone to ethical concerns.

While the study by A. Solitto Da Silva *et al.* (2023) used ontologies to detect errors in C# code, the present research employed various ML algorithms in Python, focusing on automating error detection. This provides greater flexibility when working with large datasets. Moreover, the study by L.T. Yi (2024) used the supervised ML algorithm Gaussian Processes for Java programs. However, the present research applied a broader range of ML methods for error detection in code, allowing for adaptability to different tasks and programming languages. Unlike the study by S. Himashree *et al.* (2024), which applied eXtreme Gradient Boosting for detecting undesirable parameters in clinical trials, the present approach was tailored specifically to error detection in programs. Nevertheless, the approaches of this study can be adapted to various fields, including not only the clinical domain but also other industries where automated analysis of large datasets and anomaly detection are crucial.

Whereas the study by P. Chakraborty *et al.* (2024) proposed reinforcement learning methods for error localisation in code, the present research focused on applying ML methods for error detection. The approach in the referenced study was centred on error localisation with direct metric optimisation, which can be useful in specific cases where precise determination of error location is required. However, the present approach provides greater flexibility in handling different types of errors, covering more aspects of the detection process. The study by S. Wang *et al.* (2024) focused on detecting errors in register-transfer-level code using a neural network, which operates only in highly specialised fields such as chip design. Therefore, the current results are more universal for various software projects. Additionally, in the research by M. Kumari *et al.* (2024b), a combination of entropy and ML algorithms was applied for bug prioritisation, which is similar to the present approach. However, this study not only focused on error prediction but also on their resolution, improving programmers' efficiency and reducing defect correction time.

This study demonstrated outstanding results in automated error detection in code using ML methods, allowing efficient processing of large datasets and identifying various types of errors. This provides greater flexibility and accuracy compared to traditional static tools, such as those in the study by N.S. Harzevili *et al.* (2023), where static tools detected only 0.01% of errors, highlighting the limited effectiveness of such methods in complex ML systems.

On the other hand, in the work of T. Aladics *et al*. (2021), the use of abstract syntax trees and the Doc2Vec algorithm was proposed to create vector representations of code, which improved error prediction accuracy beyond methods relying solely on code metrics. The present research complements this approach as it not only aids in prediction but also eliminates errors, making the process more comprehensive and adaptable to various tasks. Similarly, the study by L. Rehman *et al*. (2023) used statistical features for error detection, with ensemble models and neural networks showing high accuracy. In this regard, the present study confirms the effectiveness of existing models and proposes a more universal solution that integrates both error detection and correction.

The present study considered five approaches to error detection, allowing for the selection of the most effective method depending on the task. While the authors Z. Li *et al*. (2023) used only one approach – WELL, based on weakly supervised learning, which showed good results in error localisation – the present study is more flexible and covered multiple methods, enabling better adaptation to different error types. Regarding the work of K. Sarawan *et al*. (2023), their research demonstrated the effectiveness of Logistic Regression for classifying errors by severity and Long Short-Term Memory for specific datasets. However, the present study offers greater versatility and better accounts for error specifics, providing a comprehensive solution for error detection and resolution in code.

This study focused on using ML for automated software error detection. In contrast, in the work of F. Fazal & C. Farook (2024), where the ExtraTreesClassifier algorithm achieved the highest accuracy in depression detection in textual data, a method tailored to text analysis was used. Therefore, the results of the referenced study are not directly related to software error detection, but they demonstrate the effectiveness of ML algorithms for classification, partially confirming the potential of such methods for various tasks. As for the study by W. Albattah & M. Alzahrani (2024), where Long Short-Term Memory exhibited high accuracy in predicting software bugs, the current results confirmed the effectiveness of similar approaches. However, unlike the referenced study, which focused on a single method, the conducted research proposed several different approaches, allowing for a more tailored solution for various error types.

Overall, the obtained results confirm that modern ML methods have great potential for automating error detection in code, particularly due to their ability to work with large and complex datasets. This enables high accuracy and speed in error prediction, representing a significant advancement over traditional methods.

CONCLUSIONS

The research results demonstrated the high effectiveness of applying (ML) for automated error detection in code, particularly when working with large and complex data. Classification using decision trees yielded good results when analysing smaller datasets, where straightforward result interpretation and processing speed are crucial. However, this approach has limited efficiency when handling complex structures and large data volumes. Sequence analysis using recurrent neural networks (RNNs) proved effective in tasks requiring contextual and sequential dependencies, particularly for analysing program logic. However, these methods were sensitive to the vanishing gradient effect when processing long sequences, reducing their effectiveness in certain cases. In turn, clustering algorithms were useful for anomaly detection in code, especially when identifying new or previously unknown error types. However, these methods require well-tuned parameters and do not always achieve high accuracy, particularly in complex scenarios. A generative approach using transformers showed the best results in tasks involving large data volumes and complex code structures. This method effectively processes substantial amounts of information but requires significant computational resources for training and application. Moreover, deep learning with convolutional neural networks (CNNs) demonstrated good efficiency in code visualisation tasks but had limited capability in capturing sequential context.

Recommendations for further research include improving the efficiency of transformers through the integration of new architectures and methods to reduce computational costs, as well as developing approaches for automated error localisation to enhance accuracy and convenience in defect correction. Additionally, further exploration of methods for optimising classical algorithms is advisable to improve their efficiency in more complex cases.

A limitation of the study is that the examined methods have varying computational requirements, which may restrict their application in real-world environments with limited resources. Furthermore, the limited amount of test data for specific methods prevents a definitive determination of the best approach for all types of code errors.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Abed Alsaedi, S., Noaman, A.Y., Ahmed, A., & Eassa, F. (2023). Nature-based prediction model of bug reports based on ensemble machine learning model. *IEEE Access*, 11, 63916-63931. doi: 10.1109/ACCESS.2023.3288156.

[2] Aladics, T., Jász, J., & Ferenc, R. (2021). Bug prediction using source code embedding based on Doc2Vec. In O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B.O. Apduhan, A. Rocha, E. Tarantino & C.M. Torre (Eds.), *Computational science and its applications – ICCSA* 2021 (pp. 382-397). Cham: Springer. doi: 10.1007/978-3-030-87007-2_27.
[3] Albattah, W., & Alzahrani, M. (2024). Software defect prediction based on machine learning and deep learning techniques: An empirical approach. *AI*, 5(4),1743-1758. doi: 10.3390/ai5040086.
[4] Batchu, P., Gali, T.R., & Inturi, S. (2024). Python source code analysis for bug detection using transformers. *Engineering and Technology Journal*, 9(4), 3772-3777. doi: 10.47191/etj/v9i04.15.
[5] Chakraborty, P., Alfadel, M., & Nagappan, M. (2024). RLocator: Reinforcement learning for bug localization. *IEEE Transactions on Software Engineering*, 50(10), 2695-2708. doi: 10.1109/TSE.2024.3452595.
[6] Chatterjee, P., & Das, A. (2023). Leveraging machine learning for predictive bug analysis. *International Journal of Scientific Research and Management*, 12(12), 1804-1814. doi: 10.18535/ijsrm/v12i12.ec04.
[7] Chen, J., Xiang, H., Li, L., Zhang, Y., Ding, B., & Li, Q. (2024). Utilizing precise and complete code context to guide LLM in automatic false positive mitigation. *Computer Science: Software Engineering*, 1(1), article number arXiv:2411.03079. doi: 10.48550/arXiv.2411.03079.
[8] Fazal, F., & Farook, C. (2024). A machine learning approach for depression detection in Sinhala-English code-mixed. *International Journal on Advances in ICT for Emerging Regions*, 17(3), 102-112. doi: 10.4038/icter.v17i3.7282.
[9] Grytsiuk, P.Y., Sikora, L.S., & Hrytsiuk, Y.I. (2023). Problems of detecting and correcting errors in messages encoded by Fibonacci matrices. *Scientific Bulletin of UNFU*, 33(4), 45-58. doi: 10.36930/40330407.
[10] Harzevili, N.S., Shin, J., Wang, J., Wang, S., & Nagappan, N. (2023). Automatic static bug detection for machine learning libraries: Are we there yet? *Computer Science: Software Engineering*, article number arXiv:2307.04080. doi: 10.48550/arXiv.2307.04080.
[11] Himashree, S., Ajitha, S., Reshma, K.J., & Girish, R. (2024). Automating adverse event detection in clinical trials through machine learning. *Periodico di Mineralogia*, 93(6), 340-352. doi: 10.5281/zenodo.14282801.
[12] Kodamasimham, K., Murthy, P., & Sarangi, S. (2024). Exploring the synergy between generative ai and software engineering: Automating code optimization and bug fixing. *World Journal of Advanced Engineering Technology and Sciences*, 13(1), 682-691. doi: 10.30574/wjaets.2024.13.1.0464.
[13] Kumar, S.A., & Prasanna, B. (2024). Software bug prediction using machine learning. *International Journal of Scientific Research in Engineering and Management*, 8(4). doi: 10.55041/IJSREM31425
[14] Kumari, M., Singh, R., & Singh, V.B. (2024a). Prioritization of software bugs using entropy-based measures. *Journal of Software: Evolution and Process*, 37(2), article number e2742. doi: 10.1002/smr.2742.
[15] Kumari, N., Ahmed, J., Raghuwanshi, K.K., & Agarwal, P. (2024b). Automating bug triage: Unleashing the power of machine learning and natural language processing. doi: 10.21203/rs.3.rs-4294244/v1.
[16] Li, Z., Zhang, H., Jin, Z., & Li, G. (2023). WELL: Applying bug detectors to bug localization via weakly supervised learning. *Journal of Software: Evolution and Process*. doi: 10.22541/au.168810846.65018089/v1.
[17] Nadim, M., & Roy, B. (2022). Utilizing source code syntax patterns to detect bug inducing commits using machine learning models. *Software Quality Journal*, 31, 775-807. doi: 10.1007/s11219-022-09611-3.
[18] Nouwou Mindom, P.S., Da Silva, L., Nikanjam, A., & Khomh, F. (2024). Continuously learning bug locations. *Computer Science: Software Engineering*, article number arXiv:2412.11289. doi: 10.48550/arXiv.2412.11289.
[19] Pozharytska, O., & Troitskyi, K. (2021). Digital technologies for grammatical error correction: Deep learning methods & syntactic N-grams. *Issues of Translation Studies and Language Teaching Methods*, 35, 237-241. doi: 10.18524/2307-4558.2021.35.237789.
[20] Rehman, L., Iqbal, M.J., Ramzan, S., Nawaz, S., Jaffar, A., Yaqoob, S., & Ul Haq, I. (2023). Long-lived bugs prediction using machine learning approaches. *Journal of Jilin University, Engineering and Technology Edition*, 42(1), 76-97. doi: 10.17605/OSF.IO/T2Z75.
[21] Sarawan, K., Polpinij, J., & Luaphol, B. (2023). Machine learning-based methods for identifying bug severity level from bug reports. In P. Meesad, S. Sodsee, W. Jitsakul & S. Tangwannawit (Eds.), *Proceedings of the 19th international conference on computing and information technology* (pp. 199-208). Cham: Springer. doi: 10.1007/978-3-031-30474-3_17.
[22] Shah, M., Sharma, A., & Kumar, R. (2023). Software bugs detection using supervised machine learning techniques. *Advances in Science and Technology*, 124, 594-601. doi: 10.4028/p-of2831.
[23] Solitto Da Silva, A., Garcia, R.E., & Botega, L.C. (2023). Bug localization model in source code using ontologies. *IEEE Access*, 11, 98542-98557. doi: 10.1109/ACCESS.2023.3313598.
[24] Symala Sai Sree, M., Sai Deekshitha, P., Jhansi, P., & Alekya, R. (2024). Nature-based prediction model of bug reports based on ensemble machine learning model. *Journal of Engineering Sciences*, 230-240. doi: 10.36893/JES.2024.V15I10.028.

[25] Wang, S., Zheng, H., Wen, X., Xu, K., & Tan, H. (2024). Enhancing chip design verification through AI-powered bug detection in RTL code. *Applied and Computational Engineering*, 92, 27-33. doi: 10.54254/2755-2721/92/20241685.
[26] Yanko, A., Prokudin, A., Fil, I., & Kruk, O. (2024). Detection of LDDoS attacks using SDN networks with machine learning elements. *Measuring and Computing Devices in Technological Processes*, 4, 287-296. doi: 10.31891/2219-9365-2024-80-36.
[27] Yi, L.T. (2024). Code smell detection using machine learning classification algorithm. *International Journal of Research and Innovation in Social Science*, 8(5), 889-900. doi: 10.47772/IJRISS.2024.805063.

Автоматизація виявлення помилок у коді за допомогою машинного навчання

Олена Сокол

Магістр
Київський національний університет імені Тараса Шевченка
01033, вул. Володимирська, 60, м. Київ, Україна
https://orcid.org/0009-0005-5160-460X

Анотація. Мета роботи полягала в розробці підходів до автоматизованого виявлення помилок у коді шляхом використання алгоритмів машинного навчання (ML). У дослідженні розглянуто п'ять основних підходів: класифікація за допомогою дерев рішень, аналіз послідовностей з використанням рекурентних нейронних мереж, пошук аномалій через алгоритми кластеризації, генеративний підхід з трансформерами та глибоке навчання з використанням згорткових нейронних мереж. Для кожного підходу проведено оцінювання за п'ятибальною шкалою, яке базувалося на систематичному вивченні переваг і недоліків, з урахуванням показників виконання. Результати включають приклади реалізацій цих підходів, аналіз переваг та недоліків, а також оцінки їх ефективності. Трансформери продемонстрували високу точність у складних випадках, ефективно обробляючи великі обсяги даних і виявляючи помилки в складних структурах коду. Цей підхід отримав оцінку 5 балів завдяки своїй здатності до високої точності та ефективності в роботі з великими і складними даними. Алгоритми дерев рішень, незважаючи на свою швидкість і простоту, мали обмежену ефективність у масштабних задачах, особливо для складних програмних структур. В той же час, алгоритми кластеризації показали свою універсальність у виявленні аномалій, однак їх точність залежала від правильності вибору параметрів кластеризації. Ці алгоритми отримали оцінки 3 бали через обмежену ефективність у складних задачах та проблеми з масштабованістю. В свою чергу, підхід на основі рекурентних нейронних мереж продемонстрував хороші результати при аналізі послідовностей, але був чутливий до довгих послідовностей та ефекту зникання градієнта, що знижувало його точність. Згорткові нейронні мережі ефективно працювали з візуальними репрезентаціями коду, але мали обмежену здатність до врахування контексту послідовностей. Підходи на основі нейронних мереж отримали оцінки 4 бали, оскільки ефективні в специфічних задачах, але мають обмеження, пов'язані з ресурсами та контекстуальним аналізом. Таким чином, результати підтвердили, що для автоматизованого виявлення помилок у великих і складних програмах найбільш ефективним є використання генеративних моделей, таких як трансформери, що здатні обробляти значні обсяги даних з високою точністю

Ключові слова: дерева рішень; аналіз послідовностей; пошук аномалій; генеративні трансформери; алгоритми кластеризації та класифікації; використання нейронних мереж

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 004.89:004.056.5
DOI: 10.62660/bcstu/1.2025.48

Utilising large language models for automated real-time cyber threat analysis

Denys Kovalchuk*
Postgraduate Student
International Humanitarian University
65009, 33 Fontanska Doroha Str., Odesa, Ukraine
https://orcid.org/0009-0003-2302-8698

Abstract. In the contemporary cybersecurity landscape, where the rapid growth in the quantity and complexity of threats has undermined the effectiveness of traditional rule- and signature-based detection methods, an urgent need has emerged for automated cyber threat analysis systems employing large language models. The objective of this study was to investigate the capabilities of large language models for automated cyber threat analysis, risk assessment, and improving incident response efficiency in corporate environments. To achieve this goal, machine learning and natural language processing techniques were employed, particularly the adaptation of large language models for threat classification, risk-level evaluation, and anomaly detection. A system was developed to analyse incoming and outgoing email communications, which during testing automatically identified phishing attacks and social engineering techniques, assigned risk scores to messages, and quarantined those exceeding a predefined threshold (e.g., 0.8) for further inspection. The system analysed a dataset of 100,000 emails, of which 70% were legitimate communications and 30% were phishing attacks. Additionally, real-time analysis of data streams from corporate logs and external sources enabled the detection of potential cyber incidents with an accuracy of up to 94%, while reducing the false-positive rate to 6.5%. The obtained results confirmed the efficacy of large language models, which achieved a threat classification accuracy of up to 97% with an F1-score of 95% and reduced incident response times by 30-40%. These findings can be leveraged by other researchers to refine phishing detection techniques, reduce false positives in corporate security systems, and integrate machine learning models with diverse data sources, including SIEM systems and external cybersecurity resources

Keywords: natural language processing; machine learning for security; phishing attack detection; anomaly detection; deep learning in cybersecurity; neural networks for security; cyber threat intelligence

INTRODUCTION

The advancement of digital technologies and the increasing volume of transmitted data have led to a rise in the sophistication of cyber threats, complicating their timely detection and mitigation. Cyberattacks have become increasingly dynamic, employing modern evasion techniques to bypass traditional defence mechanisms, such as signature analysis and static detection rules, and are increasingly targeting critical systems. Conventional threat analysis approaches, reliant on signature-based detection or static rule sets, demonstrate limited effectiveness against novel and modified attacks. Modern threats – such as phishing campaigns, social engineering, and malware-based attacks – often evade standard security tools due to their inability to analyse contextual cues and adapt to emerging threats in real time.

For instance, S. Jamal & H. Wimmer (2023) emphasised in their research that applying transformer-based models for phishing detection in corporate email communications significantly improves

Article's History: Received: 11.09.2024; Revised: 22.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Kovalchuk, D. (2024). Utilising large language models for automated real-time cyber threat analysis. *Bulletin of Cherkasy State Technological University*, 30(1), 48-58. doi: 10.62660/bcstu/1.2025.48.

*Corresponding author

classification accuracy and reduces the number of missed attacks. O. Cherqi *et al*. (2023) proposed a methodology combining generative adversarial networks (GANs) and contrastive learning to better identify complex and rare attack types undetected by traditional methods. Consequently, there is a pressing need for intelligent cyber threat analysis methods capable of rapidly and accurately identifying suspicious activity across diverse data sources. One promising solution lies in the application of large language models (LLMs), which possess advanced capabilities in natural language processing (NLP), textual data analysis, and uncovering hidden patterns in large datasets. The use of LLMs in cybersecurity opens new avenues for automating threat analysis processes, including monitoring corporate communications for phishing attempts, analysing web content for potentially malicious resources, and aggregating threat intelligence from open and closed sources.

In the work of K. Singh *et al*. (2022), the potential of NLP techniques combined with machine learning algorithms was explored for detecting vulnerabilities in documentation and source code. The study found that integrating these methods not only identified obvious errors but also successfully flagged potential threats arising from inaccuracies in textual descriptions. The results confirmed the efficacy of this approach in threat analysis, enhancing the security of corporate information systems. A systematic review by Y. Gholami (2024) examined the prospects of leveraging LLMs for detecting latent threat patterns. It was noted that LLMs' capacity for deep contextual analysis enabled the identification of both explicit malicious indicators and subtle patterns often overlooked by traditional signature-based methods. Additionally, the analysis of heterogeneous data sources facilitated the early detection of emerging cyber threats.

The study by L. Rybalchenko & S. Ohrimenco (2024) evaluated the applicability of LLMs and other machine learning algorithms in large enterprises. It highlighted that the effectiveness of baseline models was constrained by insufficient adaptation to specific corporate cybersecurity requirements. The findings underscored the necessity for further optimisation, domain-specific data integration, and transfer learning techniques to reduce false positives and improve threat detection accuracy. In the work of O.P. Podvysotska & S.O. Nosok (2024), machine learning algorithms were applied to network traffic analysis. The study demonstrated that combining structural and semantic analysis enabled timely diagnosis of potential threats and provided actionable response recommendations. The results indicated the feasibility of integrating such methods into security information and event management (SIEM) systems to enhance their efficacy.

In the study by V.B. Mokin & M.G. Pradivliannyi (2024), innovative approaches to developing self-learning cybersecurity systems were examined through the integration of machine learning, data mining, and the "Artificial Intelligence of Things" (AIoT) concept. It was established that combining AIoT with LLMs enabled the timely detection of novel cyberattack types and enhanced system resilience against evolving adversarial tactics. However, the study also emphasised that achieving maximum efficacy required addressing challenges such as energy consumption, computational resource scalability, and the development of additional safeguards against adversarial attacks.

A. Partyka *et al*. (2024) focused on analysing cybercrime, particularly ransomware attacks, using AI models. The study demonstrated that applying these technologies enabled deep data analysis for early detection of ransomware attack signatures. The findings indicated that integrating such approaches reduced incident response times, automated decision-making processes, and significantly improved the security management of critical infrastructures. The objective of this study was to investigate the capabilities of modern LLMs in performing automated real-time cyber threat analysis and assess their practical efficacy when integrated into corporate security systems.

MATERIALS AND METHODS

The study employed a simulated large-scale organisational infrastructure with a typical corporate network architecture. Testing was conducted in the AWS Cloud, and pre-trained LLMs were accessed via REST API. To ensure representativeness, all data channels (email communications, network logs, user behaviour data) were configured to replicate the workload of a large enterprise with approximately 2,500 workstations, including 2,000 concurrently active users and 500 additional technical and service accounts. This approach provided a realistic corporate environment for effective cyber threat analysis.

To test the hypothesis that LLMs can detect phishing emails, a dataset of 100,000 emails was compiled. Approximately 70% comprised legitimate business communications, while the remaining 30% contained various phishing scenarios. This distribution simulated real-world conditions, where genuine threats are outnumbered by legitimate traffic. Raw data were cleansed of duplicates, technical headers, and mislabelled entries, then normalised into a unified text structure. To verify email categorisation as "malicious" or "benign", samples were generated using Python scripts and open-source LLMs. Phishing examples were divided into two categories: (1) simple variants (credential harvesting via spoofed links); and (2) advanced social engineering scenarios (executive impersonation, fake newsletters, etc.). The dataset was split into training and test sets, maintaining a similar ratio of benign to malicious samples.

Multiple threat identification approaches were employed to evaluate their responsiveness to novel and modified attacks. Signature-based analysis, classical machine learning, and LLMs were selected as

they best reflect contemporary trends in corporate security systems. The first approach considered was signature-based detection, which relied on predefined patterns and keywords. This method required minimal computational resources and ensured rapid execution; however, it proved vulnerable to novel formulations and evasion techniques. The second approach involved classical machine learning algorithms (logistic regression and support vector machines), which provided better generalisation capabilities compared to signature-based methods. These algorithms operated on a set of vectorised features derived from real emails containing both phishing examples and legitimate communications. While this approach required a well-labelled dataset, it enabled the detection of previously unknown threats by identifying patterns within textual data. The third toolset comprised LLMs, notably GPT-2, GPT-3, and a specially fine-tuned version of GPT-4. These models were employed for in-depth contextual analysis of emails, owing to their ability to detect even subtle deviations in style, vocabulary, and hidden indicators of social engineering. To enhance the relevance and accuracy of threat detection within a specific environment, LLMs were additionally supplemented with internal corporate data (real email correspondences, security policies).

For model generalisability assessment, 70% of the dataset (legitimate emails) served as training data, while 30% (malicious emails) formed the test set. Training samples maintained the original dataset's phishing-to-benign ratio. Signature-based methods used manually defined keywords, classical ML models trained on vectorised features, and LLMs underwent fine-tuning on internal corporate data to capture enterprise-specific linguistic patterns. Post-training, each algorithm was evaluated on unseen test data to avoid overfitting and objectively assess detection accuracy. Performance metrics included precision, recall, and F1-score, ensuring a balance between threat detection (minimising missed attacks) and false-positive reduction (avoiding misclassification of benign emails). Computations were performed on an Intel Xeon server with AWS acceleration for efficient large-scale data processing.

Concurrently, models were tested in network activity monitoring. Data sources included firewall, proxy, and authentication server logs aggregated by SIEM systems (Splunk, Elastic Stack, etc.), totalling 10 million records. A subset contained explicit attack indicators (SQL injection, brute force, port scanning, suspicious URLs), while the remainder constituted legitimate traffic. Evaluated approaches included: Rule-based SIEM correlation detectors; Random Forest; XGBoost ensembles; Domain-adapted LLMs (GPT-4). As network logs required detection of both known and novel threats, a risk-scoring mechanism was implemented. Each record was pre-processed to extract textual and metadata features (request type, access attempts, anomalous URIs, etc.). Signature-based methods matched known exploit patterns, classical ML models evaluated feature vectors, and LLMs analysed textual log fields for unusual phrases, escalation markers, or anomalous user behaviour. To maintain detector relevance against emerging attack vectors, periodic model retraining on recent incidents was implemented. Adversarial robustness was ensured via data sanitisation (removing hidden characters, distortions) and structural anomaly checks. This approach enhanced resilience against sophisticated multi-vector attacks where traditional signatures or attacker adaptations would otherwise evade detection.

RESULTS

The hypothesis regarding the capability of LLMs to detect phishing emails was tested by constructing a dataset of 100,000 emails, where 70% comprised legitimate business correspondence and 30% contained phishing attack scenarios. This approach simulated real-world conditions, where the volume of genuine threats is significantly lower than the flow of legitimate communications. Using risk assessment mechanisms, these models assigned threat levels to incoming messages and, upon exceeding a critical threshold, redirected them to quarantine for further analysis by security experts. Additionally, LLMs facilitated effective monitoring of cyber threat intelligence gathered from diverse sources, including corporate event logs, cybersecurity blogs, forums, and dark web resources. This methodology enabled real-time threat analysis, substantially improving incident response efficiency.

A critical function of LLMs was their ability to process and analyse corporate system log files, allowing for automated detection of anomalies in network activity and timely alerts to relevant security teams. E. Joshua *et al*. (2025) described the AIDTIS system, which integrates LLMs with threat intelligence sources (such as the dark web and cybersecurity blogs), providing a more comprehensive approach to analysing malicious activity and accelerating decision-making processes for blocking or isolating adversarial actions.

The presented results summarised experimental evaluations conducted to assess the potential of LLMs in automated real-time cyber threat analysis. The study focused on applying LLMs to detect phishing emails in corporate environments, assess social engineering risks, analyse anomalies in network logs, and integrate third-party threat intelligence sources (including cybersecurity blogs, forums, and dark web resources). The research aimed to determine whether models such as GPT or Bidirectional Encoder Representations from Transformers (BERT) met expectations regarding improved incident detection accuracy and accelerated response processes in enterprise settings.

The escalation of attacks targeting large enterprises, government institutions, and critical infrastructure underscored the necessity of real-time automated cyber threat analysis. Leveraging LLMs for pattern recognition in textual data, log files, email communications,

and external threat intelligence sources enabled a deeper level of threat analysis compared to traditional methods. The proposed methodology allowed for automated identification of phishing campaigns through textual analysis, detecting signs of social engineering, and evaluating individual message risk levels. Research findings confirmed that LLMs significantly reduced false-positive rates in threat detection systems due to their ability to contextualise messages and identify hidden lexico-semantic patterns characteristic of phishing and social engineering attacks.

The dataset included both trivial scenarios (e.g., password theft via spoofed links) and sophisticated social engineering tactics (e.g., impersonation of internal communications from executives, fake corporate newsletters). All emails were labelled by experts and divided into training and test sets. Multiple approaches were evaluated: rule-based detection and signatures, classical machine learning methods (logistic regression, Support Vector Machine), and LLMs (GPT-2, GPT-3, and a fine-tuned variant of GPT-4). The study revealed that even pre-trained LLMs outperformed traditional methods in classifying malicious emails. A notable advantage was their ability to detect modified phishing techniques, where attackers altered message structure, lexicon, or format to evade signature-based detection. In practice, this meant that GPT-derived models identified lexical and syntactic anomalies at a deeper level, analysing overall context rather than relying solely on isolated keywords. Testing demonstrated that LLM-based email screening achieved significantly higher accuracy, particularly against spear-phishing campaigns tailored to specific targets.

To illustrate the performance disparity between methods, key metrics were examined: accuracy, recall, precision, and F1-score. Results were aggregated in a comparative table (Table 1). Detection thresholds were standardised across models to prevent bias from varying configuration parameters. The table also documented false-positive and false-negative ratios, as both metrics were deemed critical from a security standpoint: excessive false alarms overload Security Operations Centre (SOC) analysts, while missed attacks pose direct organisational risks.

Table 1. Comparison of accuracy and F1-score for different phishing email detection methods

Method	Accuracy	Recall	Precision	F1-score
Rules/Signatures	0.84	0.76	0.82	0.79
Logistic regression	0.88	0.81	0.85	0.83
Support vector machine	0.9	0.84	0.87	0.85
GPT-2 (fine-tuned)	0.93	0.89	0.91	0.9
GPT-3 (fine-tuned)	0.95	0.92	0.93	0.92
Integrated LLM (GPT-4)	0.97	0.94	0.96	0.95

Source: compiled by the author

Notably, the GPT-4 variant achieved the highest accuracy (0.97) while maintaining an F1-score of 0.95, demonstrating a balanced trade-off between detecting genuine phishing threats (high recall) and correctly filtering benign emails (high precision). This performance was attributed to GPT-4's expanded parameter count and domain-specific fine-tuning on corporate datasets. Crucially, despite its complexity, GPT-4 adapted dynamically to evolving social engineering tactics, detecting novel patterns absent from initial signature databases.

Subsequent experiments evaluated LLM-driven "risk scoring" mechanisms for incoming emails. Each message was analysed for lexico-semantic features (keyword frequency, contextual usage), embedded links, attachments, and stylistic consistency with typical corporate correspondence. If a message's risk score exceeded a dynamic threshold (calibrated per environment), it was automatically quarantined for SOC review. This approach proved particularly effective against sophisticated attacks where legacy systems relied solely on known malicious link signatures or rigid detection rules. A model such as GPT-4 was capable of responding even to concealed and non-standard formulations, as well as detecting atypical use of communicative genre – for instance, the simulation of a scenario in which a department head addressed an employee in an uncharacteristic tone or issued a request for confidential information that would not align with normal organisational practices.

Further analysis examined LLM applications in network activity monitoring and log anomaly detection. A typical SIEM component aggregated logs from firewalls, authentication servers, email gateways, and proxies. The experiment involved a sample of 10 million log entries (user authentications, connection attempts, database queries, module errors). Experts labelled a subset as routine/"noise" and identified genuine cyberattacks (SQL injections, port scanning, low-intensity DDoS, brute-force attempts). Evaluated methods included: (1) rule-based SIEM correlation, (2) Random Forest, (3) XGBoost ensemble, and (4) integrated LLM (GPT-4 with domain adaptation). Key metrics were: mean processing time per log entry (ms), false-positive rate, and true-positive detection rate (Table 2).

Table 2. Comparison of approaches to anomaly detection in corporate logs

Approach	Average processing time (ms)	False positives (%)	Detected real threats (%)
Correlation Rules (Baseline SIEM)	0.5	15.2	83
Random Forest	1.7	12.8	88.5
XGBoost	2	10.2	90.1
Integrated LLM (GPT-4, fine-tuned)	2.5	6.5	94.3

Source: compiled by the author

During the analysis of these results, it was observed that the rule "the more complex the model, the longer the processing time" was indeed confirmed: GPT-4 required an average of 2.5 ms per log line, whereas the baseline SIEM processed data in 0.5 ms. However, a significantly lower number of false positives was detected (6.5% compared to 15.2% for correlation rules), along with a higher percentage of identified threats (94.3% vs. 83%). Consequently, there was a clear benefit in security analysts receiving fewer "empty" alerts while being more likely to capture genuine incidents. The reduction in manual verification volume, in turn, shortened the overall response time to sophisticated cyberattacks and diverted fewer human resources. Additional stress tests (10 million logs) demonstrated that the integrated LLM maintained stable efficiency without significant degradation in throughput or accuracy.

Parallel testing examined how LLM behaved when integrated with heterogeneous data sources outside the corporate environment, such as specialised blogs, forums, Cyber Threat Intelligence services, and darknet resources. It was observed that GPT-derived models quickly "picked up" emerging trends when certain vulnerabilities or exploits were frequently mentioned in forum posts or blogs. Such contextual information helped prioritise network events associated with potential exploit usage. This reduced the risk of missing attacks during the period before vendors released official security updates. As a result, Zero-Day detection became more agile, as LLMs could evaluate even indirect references to novel attacker techniques.

Additionally, analysis of embedded file content (e.g., Microsoft Office documents with macros) or links enabled the system to identify potentially malicious activity faster than conventional signature-based antivirus checks. In simulated attack scenarios where, malicious components were loaded in stages, traditional antivirus solutions often failed to detect threats until the final file or script was activated. In contrast, the LLM, by analysing message context and attachment characteristics, flagged high-risk indicators at the email ingestion stage. The ratio of misclassified "safe" emails with attachments decreased by approximately 13-15% compared to traditional methods, significantly lowering the likelihood of successful exploitation.

Furthermore, the hypothesis regarding improved incident detection and response metrics was confirmed. SOC workflows accelerated due to automated preliminary analysis, which filtered out "noisy" alerts and isolated incidents requiring expert scrutiny. The language model not only signalled potential threats but also provided concise explanations for their risk (e.g., alignment with new exploit intelligence, atypical user behaviour, or semantically suspicious text). Analysts thus received a preliminary "dossier" for each incident, enabling faster decision-making.

However, LLM integration also presented challenges. Continuous fine-tuning was necessary due to the rapidly evolving threat landscape and adversarial evasion techniques. Outdated models risked misclassifying novel attack patterns absent from their original training data. Protecting the model against adversarial inputs – where attackers deliberately manipulate text to deceive the LLM – was also critical. While GPT-4 exhibited resilience to minor lexical or syntactic alterations, carefully crafted adversarial constructs could still bypass detection. Thus, robust adversarial testing and secure training frameworks were recommended.

A key finding emphasised the benefits of integrating LLMs with other cybersecurity subsystems (firewalls, IPS/IDS, Data Loss Prevention modules). Such interoperability created a holistic data chain, correlating anomalous network activity, suspicious emails, and external threat intelligence. This "multi-layered defence" positioned the LLM as a unifying layer, transforming heterogeneous data (logs, forum texts, messages) into actionable signals, thereby reducing blind spots and improving threat correlation. Technical implementation required substantial computational resources: AWS Cloud services were utilised, with LLM operations conducted via Rest API. Though deployment costs and complexity increased, the resultant risk reduction, cost savings from mitigated incidents, and reputational gains justified the investment. Smaller enterprises could leverage cloud-based LLM services to minimise local infrastructure demands.

Overall, high-capacity LLMs like GPT-4 proved exceptionally valuable for analysing phishing emails and logs, enhancing both security and automation. Accuracy exceeded 0.95, with false positives reduced by several percentage points compared to classical methods – significantly improving SOC efficiency. Contextual analysis (semantic patterns, external threat feeds, corporate communication nuances) also made detection more adaptable, reducing reliance on static signatures or rules. This resilience was particularly effective against evolving phishing tactics and network anomalies.

Notably, incident response times improved. During testing, LLM-based pre-filtering assigned "priority levels" and explanatory context to logs/alerts, allowing SOC analysts to focus on critical incidents without sifting through false alarms. Decision-making cycles shortened by 30-40% compared to baseline signature/correlation-rule approaches. Nevertheless, safeguards against LLM-specific attacks (e.g., input manipulation or adversarial text) remained essential. Experiments documented cases where attackers obfuscated malicious intent via complex symbols or metadata, though GPT-4's contextual processing demonstrated superior robustness. Future work should integrate secure training methods and regular input audits to detect anomalous patterns.

Overall, the results confirmed the validity of applying LLMs in cybersecurity systems for automated real-time threat analysis. Improvements were observed in the detection of sophisticated phishing attacks, a significant reduction in false positive rates, and accelerated response times to potential incidents. The integration of LLMs with other components (e.g., traditional SIEM solutions, firewalls, Data Loss Prevention tools) provided an additional synergistic effect: information from heterogeneous sources was correlated at the semantic analysis level, facilitating the timely detection of non-obvious indicators of cyberattack preparation or deployment. NLP and deep learning technologies enabled the formation of a more comprehensive threat landscape, substantially enhanced the efficacy of Threat Intelligence practices, and paved the way for implementing a fully-fledged Threat Hunting approach.

The practical conclusion drawn from the obtained results emphasised the importance of regular model updates and rigorous testing to ensure sustained relevance in the rapidly evolving cybersecurity environment. It was recommended to track new LLM versions and explore their adaptability to specific sectoral requirements (industry, financial institutions, public sector, etc.). Specialised fine-tuning and periodic audits helped maintain high threat detection accuracy and prevent the accumulation of "blind spots". This approach allowed for more effective utilisation of NLP capabilities in detecting evolving and increasingly sophisticated variants of cyberattacks.

Thus, empirical evidence demonstrated that integrating LLMs into corporate defence ecosystems proved promising, enabling reduced data analysis time, improved threat classification accuracy, and a significant decrease in false positive alerts. Implementing such solutions required adequate computational resource preparation, involvement of competent machine learning specialists, and meticulous work with large datasets. At the same time, the resulting time and resource savings, alongside reduced likelihood of successful breaches, offset the implementation costs and complexity.

DISCUSSION

The study confirmed that adapting LLMs for real-time automated cyber threat analysis significantly improved phishing attack detection accuracy and accelerated incident response. Compared to predefined rule-based approaches, GPT-type models enabled contextual message analysis, detection of hidden social engineering patterns, and faster adaptation to novel threats as attackers altered their lexicon or format. These observations aligned with the work of J.T. Santoso *et al*. (2024), which highlighted the ability of transformer architectures to enhance response efficacy against complex attacks through self-attention mechanisms. The study achieved qualitative metrics even with relatively small datasets, while under substantially larger-scale conditions, the approach demonstrated robustness in more realistic scenarios.

The results also underscored the importance of multi-source analysis, where LLMs simultaneously processed corporate email content, network logs, and external threat intelligence (cybersecurity blogs, forums, darknet data). In such scenarios, the model better recognised potentially malicious activity by correlating indirect threat indicators across diverse data formats. A similar principle was outlined in O. Ogundairo & P. Broklyn (2024), where NLP algorithms were enriched with event log metadata, enabling more precise security incident classification and prioritisation. Comparative analysis showed that LLM deployment markedly reduced threat detection windows and strengthened contextual analysis.

The phishing email dataset used in the experiment included varying complexity levels: from classic credential harvesting to advanced social engineering variants mimicking inter-departmental corporate correspondence. GPT models successfully identified even spoofed executive communications, which often evaded simpler detectors. This trend corroborated the findings of P. Damola & A. Miracle (2024), who demonstrated LLMs' capability to autonomously recognise non-standard intrusion indicators and serve as a foundation for real-time suspicious activity blocking. S.M. Taghavi & F. Feyzi (2024) emphasised LLMs' effectiveness in rapid code and documentation review for detecting malicious patterns.

Substantial improvements in precision and F1-scores were achieved through domain-specific pre-training or fine-tuning. Analogous to the "SecureBERT" concept in E. Aghaei *et al*. (2022), GPT-4's competency increased when provided with representative corporate examples of real correspondence, security policies, and phishing patterns. Accuracy metrics (up to 0.97) surpassed traditional machine learning algorithms by 7-9% and signature-based approaches by up to 13%. Compared to P. Ranade *et al*. (2021), where generative transformers created synthetic phishing emails for training, the current study utilised significantly larger volumes of real samples, enhancing the model's ability to discern even subtle lexical deviations.

Concurrently, particular attention was required to protect LLMs from adversarial actions, as stressed by S. Liu *et al.* (2024). The research documented cases where attackers deliberately embedded hidden characters or distorted email structures to confuse models. While GPT-4 exhibited greater resilience to lexical obfuscation, data filters and regular red-team audits remained necessary to maintain system relevance. This approach aligned with warnings from M. Alabi & F. Ademola (2024), who analysed multiple attack vectors – from data poisoning to generative adversarial examples – and advocated for comprehensive detection mechanisms.

Prompt engineering also proved critical. GPT-4 required meticulous query calibration to ensure correct interpretation of threat patterns and their contextual relevance. This finding intersected with S. Parmar & H. Patel (2024) results, where well-formulated prompts substantially improved response accuracy. Early identification of phishing indicators mimicking internal corporate style emerged as a key success factor. M. Kuppam (2024) noted that application telemetry analysis helped detect attack-related behavioural anomalies.

Network log analysis – particularly SMTP headers, HTTP requests, authentication attempts, and other traffic elements – also showed promise. LLMs processed vast log volumes, correlated user actions with known malicious scanning/brute-force patterns, and assigned elevated risk scores where rule-based detection failed. This resonated with C. Avci *et al.* (2024) conclusions on self-attention mechanisms for complex threat detection. GPT-4 successfully identified subtle behavioural chains where multiple seemingly insignificant anomalies collectively indicated multi-stage attack preparation. N.G. Ambekar & S. Thokchom (2024) highlighted how unsupervised learning (e.g., variational autoencoders) could be augmented with LLMs for Zero-Day threat detection.

Comparative analysis with prior research revealed specific LLM limitations. Firstly, computational infrastructure demands grew substantially: GPT models (especially GPT-4) were resource-intensive, and near-real-time analysis required hardware scaling. However, cost-benefit analyses showed this was offset by reduced false positives and faster incident resolution. P. Gunda & T.R. Komati (2024) emphasised transformer flexibility and scalability, enabling deployment across sectors processing large volumes of complex textual data.

Secondly, questions arose concerning interpretability. Although GPT-4 could generally provide a concise commentary on why a particular email or log entry appeared suspicious, the decision-making mechanism remained challenging to explain in detail at the level of individual model layers. This aspect was also highlighted by F. Olaoye & A. Egon (2024), who proposed methods to enhance the transparency of artificial intelligence decision-making. During the research, a "general risk description" was employed, which nonetheless facilitated the work of SOC analysts by identifying atypical elements, though it did not always reveal the transformer's complex underlying logic.

In line with the recommendations of T. McIntosh *et al.* (2023), who utilised GPT-4 for formulating governance, risk management, and compliance policies, it was logical to apply LLMs not only for email detection but also for generating internal guidelines or "best practices" for responding to emerging threats. Within this study, the model also successfully drafted template service notifications when rapid employee alerts were required regarding new phishing campaigns. The examination of corporate system logs operating in near real-time confirmed the hypothesis regarding the advantage of LLMs in cases where attacks lacked classical signatures. For instance, attackers could fragment a malicious "chain" into a series of minor actions or periodically modify query parameters. GPT-4, owing to its in-depth semantic and contextual analysis, detected indicators of malicious activity that were difficult to formalise using SIEM correlation rules alone. A similar concept was observed in Z. Ding *et al.* (2023), where the hybridisation of BERT with Graph Convolutional Networks facilitated the detection of "hidden" patterns in code or metadata. In the conducted experiment, GPT-4 – without employing graph-based extensions – also demonstrated that contextual analysis could be key to filtering complex or obfuscated attack scenarios.

Considering approaches to prolonged attacks, LLMs such as GPT-4 also assisted in analysing compromise chains. L. Li & W. Chen (2024), for example, described how persistent threats could be better tracked when mechanisms for understanding interrelated events were in place. In this study, GPT-4, integrated with SIEM, effectively served as an additional layer that extracted potential anomalies from textual log descriptions and helped identify the "ground zero" of a compromise. This alignment with other studies reinforced the conclusion that LLMs significantly expanded the capabilities of traditional anomaly detection tools, particularly when discerning the holistic "narrative" of a threat.

Meanwhile, additional opportunities emerged in the field of prompt engineering. If "conversational" instructions were properly formulated, GPT-4 could provide in-depth message analysis or suggest initial steps for blocking malicious activity. A similar principle was employed by P. He *et al.* (2025) in multi-agent LLM-based systems, where "Red Team" attacks tested whether inter-agent communication channels could be corrupted. Although a multi-component architecture was not deployed, the practice of "red teaming" allowed for the detection of deliberate textual distortions, confirming the methodology's versatility.

R. Molleti *et al.* (2024) emphasised that multi-layered analysis involving LLM agents could substantially reduce response times to sophisticated cyberattacks by automating the processing of incoming data streams. With their assistance, atypical social engineering techniques or phishing variations were detected more

swiftly, as GPT-4 considered a broader context than mere alignment with a specific ATT&CK category. The presence of such "intelligent" real-time text analysis accelerated attack phase identification and helped prioritise incident handling.

However, when discussing risks, it is worth mentioning scenarios where adversaries could manipulate message content to "train" the model on false examples or poison input data. K. Alfarsi *et al*. (2024) addressed the topic of few-shot learning, where the model "learned" from a small number of samples, and highlighted the risk of rapid skewing if these samples were fabricated. During this study, the training dataset was periodically updated, but stringent quality checks were enforced on new phishing samples to prevent attackers from degrading system accuracy.

A separate area of interest was the use of LLMs for analysing external packages and dependencies when companies integrated third-party software into their systems. Analogous to W. Guo *et al*. (2024), who examined suspicious elements in npm or PyPI metadata, it was possible to process library descriptions, vulnerability reports, and even internal corporate documentation. Initial steps were taken in the test environment, and GPT-4 successfully flagged potential dangers in cases referencing "suspicious" repositories. The topic of extended data collection from dark web forums was discussed by K.S. Sangher *et al*. (2023), who developed a proactive cyber threat intelligence system and underscored the potential to detect mentions of novel attack schemes or planned data leaks. In this setup, GPT-4 received signals from both internal logs and specialised repositories dedicated to emerging evasion techniques.

The positive impact of LLMs also stemmed from reduced response times, as a greater number of "false" alerts were filtered out at the preliminary stage. For large organisations, this allowed SOC analysts to focus more quickly on serious incidents. Compared to older approaches relying solely on signatures or simple machine learning classifiers, GPT-4 demonstrated superior capability in distinguishing atypical vocabulary and word combinations. N.E. Chaabene *et al*. (2021), who researched anomaly detection in social networks, also demonstrated that multi-dimensional (including textual) analysis enabled earlier detection of suspicious activities, even if specific patterns had not yet been added to signature lists. Furthermore, LLMs were found to assist in formulating internal threat response policies. Finally, it is worth noting adjacent LLM use cases. J. Wang (2024) warned of the weaknesses in BERT-derived models and the risks of unauthorised access to their internal representations. Although GPT-4 had a different architecture, the risks of confidential parameter leakage and training data corruption remained relevant. This necessitated comprehensive safeguards: access controls, encrypted channels, network segmentation, and regular audits.

Thus, the obtained results confirmed the significant advantages of integrating LLMs into threat analysis and incident response systems: a detection accuracy of up to 0.97 for malicious emails, a substantial reduction in false positives, and accelerated response times. However, it remained critical to implement continuous retraining, protect the model from adversarial skewing, and monitor the quality of new samples. The synergy between GPT-4 and existing technologies (SIEM, antiviruses, multi-layered defence systems) amplified the effect, enabling timely mitigation of complex, previously undetected multi-stage attacks.

These findings correlated with the conclusions of Z. Ding *et al*. (2023) on deep contextual code analysis, L. Li & W. Chen (2024) on prolonged threat tracking, and K. Alfarsi *et al*. (2024) on few-shot approaches in malware detection. These works collectively affirmed the potential of transformer architectures in countering cyber threats while stressing that LLM deployment must be accompanied by systematic security measures. Ultimately, the study validated the relevance of LLMs in real-world corporate environments, demonstrated the feasibility of achieving high accuracy and rapid response times, and underscored the need for further refinement of methods to protect the models themselves from targeted attacks.

CONCLUSIONS

The findings of the study demonstrated that the integration of LLMs into cybersecurity systems significantly enhances the efficacy of real-time threat analysis. The primary objective of this work was to investigate the applicability of LLMs for the automated detection of cyberattacks based on the analysis of emails, network logs, and external data sources. The obtained results indicate that LLMs are capable of effectively recognising both known and novel threat variants, enabling deep contextual analysis of textual data.

A comparative analysis of traditional methods – signature-based approaches and classical machine learning algorithms (logistic regression, Support Vector Machine) – against LLM-based solutions revealed that models leveraging GPT-2, GPT-3, and particularly a customised version of GPT-4 exhibit superior classification accuracy and a lower false-positive rate. Due to their ability to analyse both structural and lexico-semantic features of messages, LLMs can detect even concealed patterns of phishing attacks and social engineering, substantially reducing the risk of overlooking critical incidents.

A key advantage of LLM deployment is their capacity for continuous fine-tuning in response to evolving adversarial tactics. Automated analysis, integrating data from corporate monitoring systems, email, and external resources (blogs, forums, the darknet), enables rapid response to emerging threats and minimises incident response times. This approach facilitates the establishment of a multi-layered defence framework, wherein

data from diverse sources are correlated to construct a comprehensive threat landscape.

However, alongside these benefits, the study also identified certain challenges. Among them is the necessity for persistent model updates and adaptations to prevent successful adversarial attacks aimed at deceiving LLMs through adversarial examples. Another critical aspect is ensuring sufficient computational resources for real-time model operation, which demands additional investment in supporting infrastructure. The results substantiate the viability of integrating LLMs into cybersecurity systems to enhance automation, reduce false positives, and accelerate cyber incident response processes.

The study established that traditional signature-based methods demonstrated a classification accuracy of 0.84 with an F1-score of 0.79, whereas classical machine learning algorithms – namely logistic regression and Support Vector Machine – yielded metrics of 0.88 (F1 = 0.83) and 0.9 (F1 = 0.85), respectively. GPT-2 and GPT-3 models, fine-tuned on corporate data, achieved accuracies of 0.93 (F1 = 0.9) and 0.95 (F1 = 0.92), while a specially adapted GPT-4 variant attained the highest performance – 0.97 accuracy with an F1-score of 0.95.

In network log analysis, a baseline SIEM-based detector processed data in 0.5 ms with a 15.2% false-positive rate and an 83% true-threat detection rate. The application of a Random Forest model and an XGBoost ensemble improved these metrics (1.7 ms/12.8%/88.5% and 2 ms/10.2%/90.1%, respectively). However, an integrated, fine-tuned GPT-4-based LLM achieved processing in 2.5 ms, reducing false positives to 6.5% while detecting genuine threats in 94.3% of cases. Future research should focus on strengthening LLM resilience against adversarial manipulation, refining adaptation mechanisms for sector-specific threats, and expanding toolkits for more flexible and granular model customisation in response to the evolving conditions of the modern cyber landscape.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Aghaei, E., Niu, X., Shadid, W., & Al-Shaer, E. (2022). SecureBERT: A domain-specific language model for cybersecurity. In F. Li, K. Liang, Z. Lin & S.K. Katsikas (Eds.), *Security and privacy in communication networks* (pp. 39-56). Cham: Springer. doi: 10.1007/978-3-031-25538-0_3.

[2] Alabi, M., & Ademola, F. (2024). *Adversarial robustness and defense mechanisms in machine learning*. Retrieved from https://www.researchgate.net/publication/383250866_Adversarial_Robustness_and_Defense_Mechanisms_in_Machine_Learning.

[3] Alfarsi, K., Rasheed, S., & Ahmad, I. (2024). Malware classification using few-shot learning approach. *Information*, 15(11), article number 722. doi: 10.3390/info15110722.

[4] Ambekar, N.G., & Thokchom, S. (2024). UL-VAE: An unsupervised learning approach for zero-day malware detection using variational autoencoder. In *Proceedings of the international conference on computational intelligence and network systems* (pp. 1-7). Dubai: IEEE. doi: 10.1109/CINS63881.2024.10864450.

[5] Avci, C., Tekinerdogan, B., & Catal, C. (2024). Design tactics for tailoring transformer architectures to cybersecurity challenges. *Cluster Computing*, 27, 9587-9613. doi: 10.1007/s10586-024-04355-0.

[6] Chaabene, N.E., Bouzeghoub, A., Guetari, R., & Ghezala, H.B. (2021). Deep learning methods for anomalies detection in social networks using multidimensional networks and multimodal data: A survey. *Multimedia Systems*, 28, 2133-2143. doi: 10.1007/s00530-020-00731-z.

[7] Cherqi, O., Moukafih, Y., Ghogho, M., & Benbrahim, H. (2023). Enhancing cyber threat identification in open-source intelligence feeds through an improved semi-supervised generative adversarial learning approach with contrastive learning. *IEEE Access*, 11, 84440-84452. doi: 10.1109/ACCESS.2023.3299604.

[8] Damola, P., & Miracle, A. (2024). LLM-based security automation: Revolutionizing threat detection and incident response. *Cybersecurity & Cybercrime*, 6(7), 9-15.

[9] Ding, Z., Xu, H., Guo, Y., Yan, L., Cui, L., & Hao, Z. (2023). Mal-Bert-GCN: Malware detection by combining Bert and GCN. In *Proceedings of the international conference on trust, security and privacy in computing and communications* (pp. 175-183). Wuhan: IEEE. doi: 10.1109/TrustCom56396.2022.00034.

[10] Gholami, Y. (2024). Large Language Models (LLMs) for cybersecurity: A systematic review. *World Journal of Advanced Engineering Technology and Sciences*, 13(1), 57-69. doi: 10.30574/wjaets.2024.13.1.0395.

[11] Gunda, P., & Komati, T.R. (2024). Integrating self-attention mechanisms for contextually relevant information in product management. *International Journal of Computational and Experimental Science and Engineering*, 10(4), 1361-1371. doi: 10.22399/ijcesen.651.

[12] Guo, W., Liu, C., Wang, L., Wu, J., Xu, Z., Huang, C., Fang, Y., & Liu, Y. (2024). PackageIntel: Leveraging large language models for automated intelligence extraction in package ecosystems. *ArXiv*. doi: 10.48550/arXiv.2409.15049.

[13] He, P., Lin, Y., Dong, S., Xu, H., Xing, Y., & Liu, H. (2025). Red-teaming LLM multi-agent systems via communication attacks. *ArXiv*. doi: 10.48550/arXiv.2502.14847.
[14] Jamal, S., & Wimmer, H. (2023). An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach. *Research Square*. doi: 10.21203/rs.3.rs-3608294/v1.
[15] Joshua, E., Do, J., & Patel, R. (2025). AI-driven threat intelligence system (AIDTIS): Leveraging large language models for automated threat research and detection development. *International Journal of Science and Research Archive*, 14(3), 270-285. doi: 10.30574/ijsra.2025.14.3.0339.
[16] Kuppam, M. (2024). Turning data telemetry into insights using application performance monitoring solutions. *FeedForward*, 3(2), 23-32.
[17] Li, L., & Chen, W. (2024). ConGraph: Advanced persistent threat detection method based on provenance graph combined with process context in cyber-physical system environment. *Electronics*, 13(5), article number 945. doi: 10.3390/electronics13050945.
[18] Liu, S., Chen, J., Ruan, S., Su, H., & Yin, Z. (2024). Exploring the robustness of decision-level through adversarial attacks on LLM-based embodied models. In J. Cai, M. Kankanhalli, B. Prabhakaran & S. Boll (Eds.), *Proceedings of the 32nd ACM international conference on multimedia* (pp. 8120-8128). New York: Association for Computing Machinery. doi: 10.1145/3664647.3680616
[19] McIntosh, T., Liu, T., Susnjak, T., Alavizadeh, H., Ng, A., Nowrozy, R., & Watters, P. (2023). Harnessing GPT-4 for generation of cybersecurity GRC policies: A focus on ransomware attack mitigation. *Computers & Security*, 134, article number 103424. doi: 10.1016/j.cose.2023.103424.
[20] Mokin, V.B., & Pradivliannyi, M.G. (2024). *Machine learning, intelligent data analysis and artificial intelligence of things*. Vinnytsia: Vinnytsia National Technical University.
[21] Molleti, R., Goje, V., Luthra, P., & Raghavan, P. (2024). Automated threat detection and response using LLM agents. *World Journal of Advanced Research and Reviews*, 24(2), 79-90. doi: 10.30574/wjarr.2024.24.2.3329.
[22] Ogundairo, O., & Broklyn, P. (2024). *Natural language processing for cybersecurity incident analysis*. Retrieved from https://easychair.org/publications/preprint/zXlC.
[23] Olaoye, F., & Egon, A. (2024). *Explainable AI for security decision making*. Retrieved from https://easychair.org/publications/preprint/TtJd.
[24] Parmar, S., & Patel, H. (2024). *Prompt engineering for large language model*. doi: 10.13140/RG.2.2.11549.93923.
[25] Partyka, A., Harasymchuk, O., Nyemkova, E., Sovyn, Y., & Dudykevych, V. (2024). Development of a method for investigating cybercrimes by type of ransomware using artificial intelligence models in the critical infrastructure Information security management system. *Social Development and Security*, 14(2), 52-63. doi: 10.33445/sds.2024.14.2.6.
[26] Podvysotska, O.P., & Nosok, S.O. (2024). Applying machine learning algorithms to detect network traffic anomalies. In *All-Ukrainian scientific and practical conference of students, postgraduates and young scientists* (pp. 288-290). Kyiv: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".
[27] Ranade, P., Piplai, A., Mittal, S., & Joshi, A. (2021). Generating fake cyber threat intelligence using transformer-based models. In *Proceedings of the international joint conference on neural networks* (pp. 1-9). Shenzhen: IEEE. doi: 10.1109/IJCNN52387.2021.9534192.
[28] Rybalchenko, L., & Ohrimenco, S. (2024). The impact of cybersecurity and crime on national security. *Philosophy, Economics and Law Review*, 4(2), 62-72. doi: 10.31733/2786-491X-2024-2-62.
[29] Sangher, K.S., Singh, A., Pandey, H.M., & Kumar, V. (2023). Towards safe cyber practices: Developing a proactive cyber-threat intelligence system for dark web forum content by identifying cybercrimes. *Information*, 14(6), article number 349. doi: 10.3390/info14060349.
[30] Santoso, J.T., Hartono, B., Silalahi, F.D., & Muthohir, M. (2024). Transformers in cybersecurity: Advancing threat detection and response through machine learning architectures. *Journal of Technology Informatics and Engineering*, 3(3), 382-396. doi: 10.51903/jtie.v3i3.211.
[31] Singh, K., Grover, S.S., & Kumar, R.K. (2022). Cyber security vulnerability detection using natural language processing. In *World AI IoT congress* (pp. 174-178). Seattle: IEEE. doi: 10.1109/AIIoT54504.2022.9817336.
[32] Taghavi, S.M., & Feyzi, F. (2024). Using large language models to better detect and handle software vulnerabilities and cyber security threats. *Research Square*. doi: 10.21203/rs.3.rs-4387414/v1.
[33] Wang, J. (2024). *Exploring vulnerabilities in BERT models*. doi: 10.20944/preprints202407.0204.v2.

Використання великих мовних моделей для автоматизованого аналізу кіберзагроз у режимі реального часу

Денис Ковальчук

Аспірант
Міжнародний гуманітарний університет
65009, вул. Фонтанська дорога, 33, м. Одеса, Україна
https://orcid.org/0009-0003-2302-8698

Анотація. У сучасному ландшафті кібербезпеки, де стрімке зростання кількості та складності загроз позначилося на ефективності традиційних методів виявлення, базованих на правилах та сигнатурах, було встановлено нагальну потребу у впровадженні автоматизованих систем аналізу кіберзагроз із застосуванням великих мовних моделей. Метою роботи було дослідити можливості великих мовних моделей для автоматизованого аналізу кіберзагроз, оцінки ризиків та підвищення ефективності реагування на інциденти в корпоративному середовищі. Для досягнення поставленої мети використовувалися методи машинного навчання та обробки природної мови, зокрема адаптація великих мовних моделей для класифікації загроз, оцінки рівня ризику та виявлення аномалій. Було розроблено систему аналізу вхідних та вихідних повідомлень електронної пошти, яка під час тестування автоматично ідентифікувала фішингові атаки та техніки соціальної інженерії, присвоювала повідомленням ризиковий бал і при перевищенні порогового значення (наприклад, 0.8) направляла їх у карантин для подальшої перевірки. Система аналізувала датасет із 100 000 електронних листів, з яких 70 % становили безпечні повідомлення, а 30 % – фішингові атаки. Крім того, здійснювався аналіз потоків даних із корпоративних логів та зовнішніх джерел, що дозволяло виявити потенційні кіберінциденти з точністю до 94 % та знизити відсоток хибнопозитивних спрацьовувань до 6,5 %. Отримані результати підтвердили ефективність застосування великих мовних моделей, які забезпечували точність класифікації загроз до 97 % із F1-мірою до 95 % і скорочували час реагування на інциденти на 30-40 %. Отримані результати можуть бути використані іншими дослідниками для покращення методик виявлення фішингових атак, зниження кількості помилкових спрацьовувань у корпоративних системах безпеки та інтеграції моделей машинного навчання з різними джерелами даних, включаючи SIEM-системи та зовнішні ресурси з кібербезпеки

Ключові слова: обробка природної мови; машинне навчання для безпеки; виявлення фішингових атак; виявлення аномалій; глибоке навчання у кібербезпеці; нейронні мережі для безпеки; розвідка кіберзагроз

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 025.4:004.65
DOI: 10.62660/bcstu/1.2025.59

Factors complicating the identification and processing of duplicates in bibliographic records: A theoretical perspective

Oleh Vasylenko*
Postgraduate Student
V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine
03187, 40 Academician Glushkov Ave., Kyiv, Ukraine
Lecturer
Ukrainian Institute of Arts and Sciences
08292, 14 Instytutska Str., Bucha, Ukraine
https://orcid.org/0000-0001-8498-2950

Abstract. This article examined the factors that create challenges in the process of identifying and processing duplicates in bibliographic records, which are a crucial component of the information systems of libraries, archives, and publishers. The study explored issues arising from typographical errors, variations in transliteration, the use of special characters, homoglyphs, differing word abbreviation rules, inconsistencies in author name spellings, and shortcomings in the application of standard identifiers such as ISBN and ISSN. Particular attention was given to the impact of discrepancies between international and local MARC standards – including Unimarc and Marc21 – on the creation and processing of bibliographic data. The analysis demonstrated that improper handling of bibliographic records can lead to degraded information retrieval quality for users, inaccuracies in source citations, and increased time expenditures for cataloguing and indexing. Furthermore, inconsistencies in standards impair the management of bibliographic data in multinational systems. The article also examined the consequences of these issues for bibliographic systems, including reduced search query accuracy, difficulties in data integration across catalogues, and increased time and resource costs for record processing. A set of solutions was proposed, including the adoption of unified record standards, the implementation of advanced adaptive search algorithms that account for linguistic and technical discrepancies, and enhanced authority control in bibliographic record creation. The findings have practical implications for information system developers, cataloguers, and library professionals, as they contribute to improving bibliographic databases, reducing duplicate records, and enhancing information retrieval quality for end users

Keywords: data processing; transliteration; library systems; information technology; authority control; cataloguing

INTRODUCTION

The identification and processing of duplicates in bibliographic records constitute a critical yet complex task in modern library science, information management, and digital technologies. The growing volume of information accompanying the development of electronic resources and databases presents researchers and practitioners with new challenges in ensuring the accuracy, reliability, and accessibility of bibliographic data. The problem is exacerbated by technical, organisational, and conceptual factors that lead to a significant number of duplicates and their improper handling. This, in turn, affects information retrieval quality,

Article's History: Received: 9.10.2024; Revised: 29.02.2025; Accepted: 17.03.2025

Suggested Citation:
Vasylenko, O. (2025). Factors complicating the identification and processing of duplicates in bibliographic records: A theoretical perspective. *Bulletin of Cherkasy State Technological University*, 30(1), 59-67. doi: 10.62660/bcstu/1.2025.59.

*Corresponding author

the efficiency of working with bibliographic records, and the overall performance of information systems. Contemporary research in this field highlights the necessity of a systematic approach to studying this issue, incorporating both theoretical foundations and practical solutions. Analysing the key factors influencing the identification and processing of duplicates is essential for developing effective algorithms, methodologies, and standards. This issue remains relevant not only for library systems but also for other domains dealing with large volumes of information, including publishing, education, and scientific research.

Recent studies on the identification and processing of duplicate bibliographic records focus on resolving complex issues related to metadata accuracy, data processing algorithms, and bibliographic resource integration. This review examines key publications that provide theoretical frameworks and propose practical solutions to address this problem. One seminal work is that of W. Walters & E. Wilder (2023), which investigates issues related to errors in bibliographic citations and the need to improve metadata accuracy in digital bibliographic records. Their approach emphasises the importance of implementing new data processing algorithms. Furthermore, the study proposes novel methods for detecting and eliminating duplicates in large databases, thereby enhancing the precision of bibliographic searches.

A significant portion of research concentrates on specific aspects of duplicate processing. For instance, S. McKeown & Z. Mir (2021) explore the effectiveness of various deduplication methods in systematic reviews, while B. Hammer *et al.* (2023) provide evidence for the necessity of duplicate removal to ensure the quality of scholarly works. Another key study by N. Borissov *et al.* (2022) describes automated algorithms that can facilitate this process, particularly in medical research. Data inconsistencies and duplication are primary factors negatively affecting the quality of bibliographic information. Similarly, H. Levenson *et al.* (2024) analyse how imperfections in bibliographic records can lead to losses in monograph programmes. From an algorithmic perspective, the work of B. Berger *et al.* (2021) advocates the use of Levenshtein distance for identifying similar entries in biological databases – a method that could be adapted for bibliographic applications. The publication by M. Guo *et al.* (2023) outlines strategies for data cleansing in real-world scenarios, emphasising the use of automated processes and standardised workflows. Other authors, such as C. Lunny *et al.* (2021), offer practical recommendations for managing duplicate research findings in systematic reviews. E. Benevento *et al.* (2022) investigate interactive data discovery processes in business environments, which could be applied to library systems. Meanwhile, P. Raja & K. Thangavel (2020) propose machine learning techniques for imputing missing data, which may prove useful in duplicate identification.

The objective of this study was to analyse the factors complicating the identification and processing of duplicates in bibliographic records, identify key challenges, and outline potential solutions within the context of modern digital technologies. The research employed analytical and comparative methods to uncover the peculiarities of duplicate occurrence in bibliographic systems. The informational basis comprised contemporary publications in library science and scholarly works addressing cataloguing and bibliographic record processing in digital environments. Special attention was given to analysing international standards such as MARC21 and their adaptation to Ukrainian library contexts.

Identification and Processing of Duplicates in Bibliographic Records

The primary challenge researchers face in identifying duplicates is the large volume of data and the complexity of processing varying record formats. As noted by P. Lizunov *et al.* (2021), ensuring the accuracy of bibliographic records requires the application of combined methods capable of identifying near-identical entries, particularly those with minor discrepancies in metadata or formatting. Their research highlights the importance of employing modern algorithms for detecting and processing duplicates, significantly enhancing search efficiency in electronic academic databases. The issue of data cleansing is also an integral part of duplicate processing. M. Gesicho *et al.* (2020) examine the process of cleaning medical data and propose strategies to improve data processing quality. Although their expertise pertains to medical systems, similar approaches can be applied to bibliographic records. Data purity is a necessary condition for ensuring the reliability of search results and the analysis of scientific information.

The analysis of bibliographic record editing issues in electronic catalogues of Ukrainian libraries is a crucial aspect of contemporary library science. O. Bruy (2015) and T. Bykova (2016) examine the challenges of editing bibliographic records in electronic catalogues, using Ukrainian libraries as a case study, and emphasise the difficulties in adapting international standards such as MARC21. Innovative methods based on artificial intelligence (AI) can also contribute to resolving the duplicate problem. R. Chauhan *et al.* (2023) describe the use of AI for detecting duplicates in software repositories, significantly reducing errors in identifying duplicate reports. However, integrating AI into infrastructure monitoring systems demonstrates considerable potential while simultaneously raising challenges regarding accuracy, ethics, and data interpretation (Alves *et al.,* 2024). Consequently, establishing ethical guidelines for AI usage becomes essential, particularly through the development of "ecological" impact declarations for sustainable development (Bogani *et al.,* 2023). These technologies can be adapted for

bibliographic records by employing specialised algorithms for duplicate detection, thereby improving the accuracy of large-scale database processing.

Furthermore, in software engineering, the use of word vector representations facilitates more precise identification of duplicate bug reports, enhancing performance in IT systems (Budhiraja *et al.,* 2018). The issue of visual similarity between Unicode characters is particularly critical for cataloguing and security systems, as homoglyphs can be exploited for deception (Coffman, 2021). Online services such as AMP-What enable the detection of duplicate characters, which is especially relevant for digital libraries (Discover your character: C, n.d.). Classical cataloguing principles remain foundational for constructing logical information structures and remain pertinent in the context of digital transformation in library science (Gorman, 2004). Another significant aspect is the application of modern methods to enhance metadata management, such as frameworks for improving information systems. C. Bockel-Rickermann *et al.* (2023) note that the use of such approaches in financial systems could be beneficial for developing analogous solutions for bibliographic records. This includes devising methods for automated metadata verification and correction, minimising human error and improving the efficiency of bibliographic systems. In software bug duplicate detection, effective feature extraction plays a key role in enhancing the accuracy of routing systems (Neysiani *et al.,* 2020).

Thanks to technological advancements, the automation of duplicate identification has become more accessible. E. Negro-Calduch *et al.* (2021) highlight the necessity of employing cutting-edge technologies to optimise electronic healthcare systems, which could also benefit library information systems. The use of methods such as data cleansing and AI algorithms significantly reduces error rates and enhances record processing quality. Duplicate identification in bibliographic systems requires a comprehensive approach that incorporates technological innovations, new data cleansing and validation methods, and the application of modern standards for integrating diverse information types. This ensures data accuracy and reliability, which are essential for effective search and management of bibliographic records. One of the key tasks is developing models that reduce errors associated with duplicates, necessitating advanced algorithms capable of processing large datasets. S. Sehra *et al.* (2020) provide an example of ensemble models for detecting duplicates in software bug reports. This study underscores the importance of combined approaches to improving duplicate detection accuracy in large-scale data systems. These models can be adapted for bibliographic records, as their underlying principles – particularly metadata classification and analysis methods – are universal and can be tailored to the needs of academic databases.

Additionally, T. Thwel & G. Sinha (2021) note that data deduplication approaches are becoming increasingly vital in the face of rapidly growing information volumes. They outline key strategies and concepts used to combat duplicates, including machine learning methods actively applied across various domains, including bibliographic record processing. Their work provides a detailed analysis of deduplication aspects that could inform the development of duplicate identification methods in bibliographic systems. Another critical dimension of the duplicate problem is the cost of processing "dirty" data. As L. Page (2005) notes, data errors can significantly increase processing costs and degrade the quality of information systems. This further underscores the need for implementing efficient algorithms to process bibliographic records and ensure their reliability. Overall, addressing the issue of duplicates in bibliographic records requires the development of new models that account for both technological innovations and the specificities of metadata processing in particular domains. These models should incorporate duplicate detection methods adaptable to modern data processing and search requirements in bibliographic systems.

Factors Complicating the Detection and Processing of Duplicates in Bibliographic Records

One of the key challenges in bibliographic record management is ensuring their uniqueness, which requires effective detection and elimination of duplicates. However, certain factors significantly complicate this process, reducing the accuracy of algorithms and automated data processing tools. Extraneous characters often go unnoticed during manual data entry or when importing records from different systems. For instance, an extra space at the beginning or end of a text string may be invisible to the user but is accounted for in automated string comparisons, causing identical records to be recognised as distinct (Hillmann, 2004).

The problem is exacerbated when multiple spaces are inserted between words or when punctuation marks are misplaced. For example, a book title may be recorded as "Bibliography: Fundamentals" or "Bibliography: Fundamentals", leading to mismatches in duplicate detection. To mitigate this issue, data-cleaning algorithms must be applied to remove extraneous characters prior to comparison (Lobuzina, 2013). When working with bibliographic records, the use of special characters – such as non-breaking spaces, tabulation marks, or non-standard encodings – also introduces complications. For instance, a non-breaking space, sometimes used to enhance text aesthetics, may not be supported by duplicate detection algorithms. As a result, even seemingly identical records may be treated as distinct due to differences in whitespace characters (Tillett, 1989). Furthermore, software such as MarcEdit may generate errors during MARC file validation if records contain tabulation marks or other invalid characters. This not only complicates file processing but also necessitates additional time for corrections.

Figure 1 illustrates an error in MarcEdit resulting from the use of tabulation.

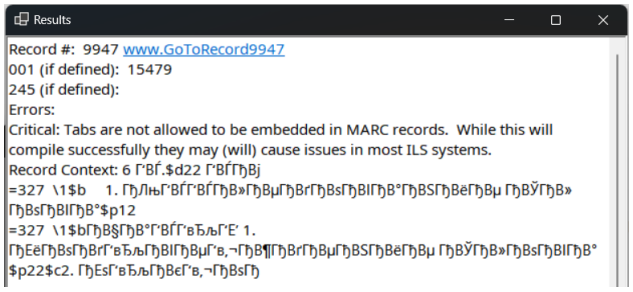

Figure 1. Error in MarcEdit due to tabulation
Source: compiled by the author

Typographical errors remain one of the most prevalent factors affecting the quality of bibliographic records. They may arise due to human error during data entry or imperfections in optical character recognition (OCR) tools. For example, misspelling an author's name, book title, or publication year may prevent duplicate identification or lead to incorrect merging. To minimise the impact of typographical errors, it is essential to implement spell-checking algorithms, machine learning techniques, and tools that compare records against authoritative databases. In multilingual bibliographic records, special characters are often replaced with visually similar alternatives or combinations. For example:

1. The letter Å may be replaced with A;
2. The symbol © may be rendered as (c);
3. The letter ß may be substituted with ss;
4. Fractions such as ½, may be written as 1/2.

These transformations are typically associated with several factors: the use of outdated character encodings, such as ASCII instead of Unicode, which supports a broader range of characters; a lack of knowledge or technical capacity to input special characters by cataloguers; or deliberate optimisation for compatibility with legacy search engines incapable of processing specific symbols correctly. While this approach may facilitate text processing, it complicates the accurate detection of duplicates, as a record containing the character "ß" and another in which it is replaced with "ss" would be treated as different entries. Homoglyphs are characters that appear visually identical but have different encodings. For instance, Cyrillic and Latin letters such as "c", "a", or "o" may look the same but are encoded differently in digital systems (Hammer, 2023). This introduces additional complexity in record comparison, as such characters are considered incompatible by search algorithms (Hopkinson, 2008).

Such errors may occur both during manual text editing and in the course of automated recognition of scanned documents. For example, the substitution of the Cyrillic character "і" with the Latin "l", or confusion involving apostrophes – such as the difference between the characters ' and ‘ or their omission altogether (Walters & Wilder, 2023). To mitigate these issues, it is necessary to implement algorithms that account for potential homoglyphs and to expand validation tools to detect and correct such instances in bibliographic records. These factors necessitate a meticulous approach to data entry standardisation and the adoption of contemporary technologies to improve the quality of bibliographic data processing. Table 1 presents examples of simplified and original formats of characters used in search algorithms.

Table 1. Simplified and original character formats

Simplified	Original character
a	a ą ą ä à á ą
c	c ć ċ
d	d ď
e	e ę ė é è
g	ġ ğ
h	h
i	i í ï
j	j ȷ
k	ĸ
l	l ļ
n	n
o	o o o ȯ ọ ỏ ơ ó ò ö
p	p
q	q
s	ş
u	ʊ u ü ú ù
v	ν v
x	x ҳ
y	y ý
z	ʒ ż

Source: compiled by the author

Homoglyphs of the letter C (Fig. 2), such as characters from other alphabets or graphically similar symbols (e.g., the Cyrillic letter "С" or the mathematical multiplication sign ×), constitute one of the factors complicating the identification and processing of duplicates in bibliographic records. They may lead to errors in search or record creation, as systems often fail to distinguish their differing semantic and graphical meanings. To address this issue, it is crucial to implement character normalisation technologies and data identity verification, which will reduce the number of false duplicates and enhance the accuracy of bibliographic databases.

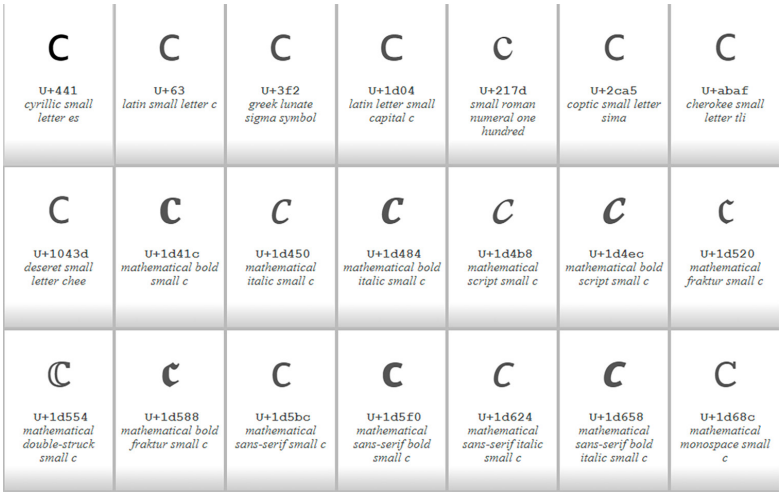

Figure 2. Homoglyphs of the letter C

Source: compiled by the author

The use of varying character cases, such as NASA and Nasa, typically does not affect duplicate detection, as most search algorithms identify such records as identical. However, when merging such records, an additional algorithm for selecting the correct spelling variant becomes necessary. This issue is particularly relevant in bibliographic records requiring precise representation of organisational names or other terms, where even minor case differences may carry significance (Borissov *et al.,* 2022). For instance, in organisational names, case usage may reflect official styling. Abbreviations are common in bibliographic records, yet their inconsistency leads to confusion. The same term may be abbreviated in multiple ways, creating potential difficulties in search and duplicate matching. For example, truncation-based abbreviations: інститут → інст.; or hyphen-based abbreviations: інститут → ін-*т* (McKeown & Mir, 2021). The more abbreviated a word is, the higher the likelihood of ambiguity, which may hinder contextual understanding and the identification of identical records. This is particularly problematic for bibliographic data reliant on the precise rendering of terms. The formulation of headings for authors and organisations varies depending on linguistic, cultural, or even institutional conventions. This may introduce additional challenges (Table 2):

Table 2. Simplified and original character formats

Difficulty	Description	Consequences
Transposed given name and surname	For example, Gresham Machen instead of Machen Gresham. Such variations complicate automated duplicate detection.	Hindered automated duplicate identification, increased errors in bibliographic records.
Name spelling variations	For instance, Smith і Smyth, or Ellen and Olena. Such discrepancies often arise from cross-linguistic conventions, alternative name forms, or typographical errors.	Increased duplicate entries due to name spelling mismatches.
Translation variants	Proper names may be rendered differently, e.g: Longbottom → Лонґботом, Долгопупс or Длиннопопп. Similarly Vyhmeister → Вихмейстер, Ваймейстер.	Search ambiguity, necessitating additional resources for cross-verification.

Source: compiled by the author

The absence of unified rules for recording identifiers is a significant issue. For instance, instead of *Taras Hryhorovych Shevchenko,* a record might only contain initials (*T. Shevchenko*), complicating record identification. Occasionally, only the first author or the responsible person is indicated, which limits search capabilities. The implementation of authority records (normative control) enables the standardisation of names, subject headings, and other terms used in bibliographic data. This not only facilitates searching and

reduces duplication but also enhances the accuracy and consistency of record creation. Standardising data entry, developing automated processing tools, and widespread adoption of normative control are key steps in addressing these challenges and improving the efficiency of working with bibliographic records.

Figure 3 illustrates a subset of author name variations recorded in the International Standard Name Identifier (ISNI) authority file. These variations demonstrate how differing transliteration approaches, linguistic discrepancies, and typographical errors affect the accuracy of bibliographic records. The lack of a unified standard for recording author names significantly complicates identification and duplicate processing. To mitigate this issue, it is advisable to employ authority files such as ISNI and implement algorithms that account for multilingualism and name variation. In gene data clustering methods, the use of fuzzy logic demonstrates increased robustness to data variability (Jahan & Hasan, 2021).

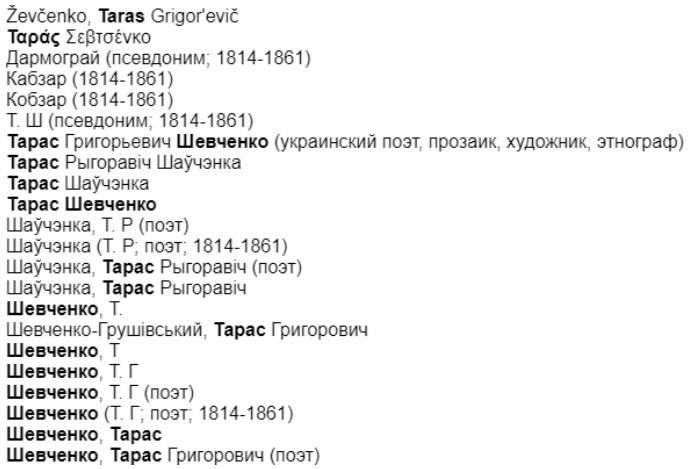

Figure 3. A subset of author name variations in the ISNI international authority file
Source: compiled by the author

Additionally, the use of different transliteration standards poses significant challenges for automated identification systems. For example, surname transliteration under the ISO standard may substantially differ from modern Ukrainian passport transliteration. This complicates the search process, particularly in multilingual databases where numerous spelling variants of the same name may exist. The issue is further exacerbated by the fact that for some languages, transliteration may only be normative in specific domains (e.g., passport records) without accommodating other specialised needs, such as library cataloguing or academic databases.

Different organisations or specialists may adopt distinct approaches to completing bibliographic records, influencing data structure and content. Consequently, information organisation structure remains pivotal in creating accessible and functional information systems (Joudrey & Taylor, 2017). For instance, some systems mandate all fields, including responsibility statements, whereas others only list the primary author or work title. Discrepancies in filling fields such as place of publication or publication year hinder record identification. Standardising these requirements would enhance the accuracy of automated data comparison. Even standard identifiers such as ISBN or ISSN do not always guarantee correct identification. For example, the same ISBN may be used for different works, particularly when publishers violate standards. The absence of an ISBN or ISSN in a record precludes automated verification. Duplicate searches often disregard information in linking fields, such as:

1. Unimarc 410 (*Series*), 411 (*Subseries*);
2. 461 (*Set*), 462 (*Subset*);
3. 482 (*Bound with...*), 454 (*Translated from...*).

Ignoring these fields may result in erroneous searches and record duplication. Another common issue arises when the primary ISBN field (e.g., 010 for Unimarc or 020 for MARC21) specifies: the ISBN of the original work instead of the translation; the ISBN of a book set instead of an individual volume. Such errors complicate precise record identification, as the system compares entries containing different data types. Depending on the standard, ISBNs may be recorded with hyphens (per DSTU and Unimarc) or without (MARC21). Consequently, the same number may be interpreted as two distinct records. In some cases, the ISBN field may contain additional information, such as clarifications in parentheses (ISBN 978-3-16-148410-0 (gift set)). Such practices violate standards and hinder automated analysis. These factors represent only a subset of issues affecting bibliographic record quality. Addressing them requires implementing data entry standards, developing effective automated processing tools, and enforcing quality control for bibliographic information.

CONCLUSIONS

To effectively resolve these issues, it is necessary to adopt standardised approaches to bibliographic record

completion, develop algorithms accounting for diverse standards, and refine normative control. The problem of bibliographic record duplication is systemic, stemming from technical, linguistic, cultural, and standardisation discrepancies. Primary negative factors include typographical errors, varying transliteration rules, lack of field completion standards, and unreliable identifiers such as ISBN. Thus, resolving these challenges necessitates implementing international standards for record unification, particularly MARC21 and IFLA LRM. The IFLA Library Reference Model (IFLA LRM) provides a conceptual framework for representing and accessing bibliographic information, aligning with modern information environment demands. More flexible search algorithms capable of accommodating multiple data representation variants should be developed. Normative control over bibliographic record creation must be reinforced, and artificial intelligence technologies for duplicate identification and processing should be advanced.

Future research prospects in this domain include the analysis of the effectiveness of existing duplicate detection algorithms across various bibliographic databases; the development of new record unification methodologies that account for multilingualism and diverse cultural contexts; the exploration of artificial intelligence integration for the automatic correction of typographical errors and transliteration; and a comparative analysis of the use of different identifiers, such as ISBN, DOI, and others, to enhance record accuracy.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Alves, V.H.M., Alves, V.A.M., & Cury, A.A. (2024). Artificial intelligence-driven structural health monitoring: Challenges, progress, and applications. In *New advances in soft computing in civil engineering. Studies in systems, decision and control* (pp. 149-166). Cham: Springer. doi: 10.1007/978-3-031-65976-8_7.

[2] Benevento, E., Aloini, D., & van der Aalst, W.M.P. (2022). How can interactive process discovery address data quality issues in real business settings? Evidence from a case study in healthcare. *Journal of Biomedical Informatics,* 130, article number 104083. doi: 10.1016/j.jbi.2022.104083.

[3] Berger, B., Waterman, M., & Yu, Y. (2021). Levenshtein distance, sequence comparison and biological database search. *IEEE Transactions on Information Theory*, 67(6), 3287-3294. doi: 10.1109/TIT.2020.2996543.

[4] Bockel-Rickermann, C., Verdonck, T., & Verbeke, W. (2023). Fraud analytics: A decade of research: Organizing challenges and solutions in the field. *Expert Systems with Applications*, 232, article number 120605. doi: 10.1016/j.eswa.2023.120605.

[5] Bogani, R., Theodorou, A., & Arnaboldi, L. (2023). Garbage in, toxic data out: A proposal for ethical artificial intelligence sustainability impact statements. *AI and Ethics,* 3, 1135-1142. doi: 10.1007/s43681-022-00221-0.

[6] Borissov, N., Haas, Q., Minder, B., Kopp-Heim, D., von Gernler, M., Janka, H., Teodoro, D., & Amini, P. (2022). Reducing systematic review burden using Deduklick: A novel, automated, reliable, and explainable deduplication algorithm to foster medical research. *Systematic Reviews*, 11, article number 172. doi: 10.1186/s13643-022-02045-9.

[7] Bruy, O. (2015). Marc21 format for the authoritative data - methodical elaboration and implantation in the library of NaUKMA. *Scientific Notes*, 22(1), 129-133.

[8] Budhiraja, A., Dutta, K., Shrivastava, M., & Reddy, R. (2018). Towards word embeddings for improved duplicate bug report retrieval in software repositories. In *Proceedings of the 2018 ACM SIGIR international conference on theory of information retrieval* (pp. 167-170). New York: Association for Computing Machinery. doi: 10.1145/3234944.3234949.

[9] Bykova, T.M. (2016). Specifics of editing bibliographic entries in the electronic catalogue: From experience of the Scientific Library of Odesa I.I. Mechnikov National University. *Bulletin of Odessa National University. Library Science, Bibliography, Book Studies*, 21(1), 137-149.

[10] Chauhan, R., Sharma, S., & Goyal, A. (2023). DENATURE: Duplicate detection and type identification in open source bug repositories. *International Journal of System Assurance Engineering and Management*, 14, 275-292. doi: 10.1007/s13198-023-01855-x.

[11] Coffman, A. (2021). *Unicode character look-alikes.* Retrieved from https://gist.github.com/StevenACoffman/a5f6f682d94e38ed804182dc2693ed4b.

[12] Discover your character: C. (n. d.). Retrieved from https://www.amp-what.com/unicode/search/c.

[13] Gesicho, M., Were, M., & Babic, A. (2020). Data cleaning process for HIV-indicator data extracted from DHIS2 national reporting system: A case study of Kenya. *BMC Medical Informatics and Decision Making,* 20, article number 293. doi: 10.1186/s12911-020-01315-7.

[14] Gorman, M. (2004). Categories and normativity. In *Categories: Historical and systematic essays* (pp. 1-22). Washington: Catholic University of America Press.

[15] Guo, M., Wang, Y., Yang, Q., Li, R., Zhao, Y., Li, C., Zhu, M., Cui, Y., Jiang, X., Sheng, S., Li, Q., & Gao, R. (2023). Normal workflow and key strategies for data cleaning toward real-world data: Viewpoint. *Interactive Journal of Medical Research*, 12, article number e44310. doi: 10.2196/44310.
[16] Hammer, B., Virgili, E., & Bilotta, F. (2023). Evidence-based literature review: De-duplication a cornerstone for quality. *World Journal of Methodology*, 13(5), 390-398. doi: 10.5662/wjm.v13.i5.390.
[17] Hillmann, D. (2004). *Metadata in practice*. Chicago: ALA Editions.
[18] Hopkinson, A. (2008). *Unimarc manual: Bibliographic format*. Munich: Walter de Gruyter.
[19] IFLA Library Reference Model (IFLA LRM). (2017). Retrieved from https://www.ifla.org/news/ifla-library-reference-model-lrm-march-2017-version-available/.
[20] Jahan, M., & Hasan, M. (2021). A robust fuzzy approach for gene expression data clustering. *Soft Computing*, 25, 14583-14596. doi: 10.1007/s00500-021-06397-7.
[21] Joudrey, D.N., & Taylor, A.G. (2017). *The organization of information*. Santa Barbara: Libraries Unlimited.
[22] Levenson, H.N., Amato, S., Bogus, I., Brody, F.E., Miller, M., & Nadal, L. (2024). Assessing bibliographic inaccuracy as a contributing factor for unintended loss in shared print monograph programs. *College & Research Libraries (C&RL)*, 85(7), article number 1021. doi: 10.5860/crl.85.7.1021.
[23] Lizunov, P., Biloshchytskyi, A., Kuchansky, A., Andrashko, Y., Biloshchytska, S., & Serbin, O. (2021). Development of the combined method of identification of near duplicates in electronic scientific works. *Eastern-European Journal of Enterprise Technologies*, 4(4(112)), 57-63. doi: 10.15587/1729-4061.2021.238318.
[24] Lobuzina, K.V. (2013). Problems of creating comprehensive digital resources of historical, cultural and scientific heritage. *Special Historical Disciplines: Issues of Theory and Methodology*, 21, 121-129.
[25] Lunny, C., Pieper, D., Thabet, P., & Kanji, S. (2021). Managing overlap of primary study results across systematic reviews: Practical considerations for authors of overviews of reviews. *BMC Medical Research Methodology,* 21, article number 140. doi: 10.1186/s12874-021-01269-y.
[26] McKeown, S., & Mir, Z.M. (2021). Considerations for conducting systematic reviews: Evaluating the performance of different methods for de-duplicating references. *Systematic Reviews*, 10, article number 38. doi: 10.1186/s13643-021-01583-y.
[27] Negro-Calduch, E., Azzopardi-Muscat, N., Krishnamurthy, R.S., & Novillo-Ortiz, D. (2021). Technological progress in electronic health record system optimization: Systematic review of systematic literature reviews. *International Journal of Medical Informatics*, 152, article number 104507. doi: 10.1016/j.ijmedinf.2021.104507.
[28] Neysiani, S.B., Babamir, S.M., & Aritsugi, M. (2020). Efficient feature extraction model for validation performance improvement of duplicate bug report detection in software bug triage systems. *Information and Software Technology,* 126, article number 106344. doi: 10.1016/j.infsof.2020.106344.
[29] Page, L. (2005). The high cost of dirty data. *Materials Management in Health Care,* 14(11), 22-25.
[30] Raja, P., & Thangavel, K. (2020). Missing value imputation using unsupervised machine learning techniques. *Soft Computing*, 24(6), 4361-4392. doi: 10.1007/s00500-019-04199-6.
[31] Sehra, S.S., Abdou, T., Başar, A., & Sehra, S.K. (2020). Amalgamated models for detecting duplicate bug reports. In *Advances in artificial intelligence* (pp. 470-482). Cham: Springer. doi: 10.1007/978-3-030-47358-7_49.
[32] Thwel, T.T., & Sinha, G. (2021). *Data deduplication approaches: Concepts, strategies, and challenges*. Amsterdam: Elsevier Inc. doi: 10.1016/C2020-0-00104-0.
[33] Tillett, B.B. (1989). *Authority control in the online environment: Considerations and practices*. Santa Barbara: Libraries Unlimited.
[34] Walters, W., & Wilder, E. (2023). Fabrication and errors in the bibliographic citations generated by ChatGPT. *Scientific Reports*, 13, article number 14045. doi: 10.1038/s41598-023-41032-5.

Фактори ускладнення ідентифікації та обробки дублікатів у бібліографічних записах: теоретичний аспект

Олег Василенко

Аспірант
Інститут кібернетики імені В. М. Глушкова Національної академії наук України
03187, просп. Академіка Глушкова, 40, м. Київ, Україна
Викладач
Український гуманітарний інститут
08292, вул. Інститутська, 14, м. Буча, Україна
https://orcid.org/0000-0001-8498-2950

Анотація. У статті було розглянуто фактори, що створюють труднощі в процесі ідентифікації та обробки дублікатів у бібліографічних записах, які є важливою складовою інформаційних систем бібліотек, архівів та видавництв. Досліджено проблеми, спричинені друкарськими помилками, різними варіаціями транслітерації, використанням специфічних символів, омогліфами, різними правилами скорочення слів, варіаціями у написанні імен авторів, а також недоліками у використанні стандартних ідентифікаторів, таких як ISBN і ISSN. Особливу увагу було приділено впливу відмінностей у міжнародних та локальних стандартах MARC, Unimarc і Marc21 на створення та обробку бібліографічних даних. Проаналізовано, що неправильна обробка бібліографічних записів може призвести до зниження якості пошуку інформації користувачами, неможливості коректного цитування джерел та збільшення витрат часу на каталогізацію та індексацію. Більше того погіршується робота з бібліографічними даними у мультинаціональних системах через розбіжності в стандартах. У статті також проаналізовано наслідки цих проблем для функціонування бібліографічних систем, зокрема зниження точності пошукових запитів, ускладнення інтеграції даних між різними каталогами та підвищення витрат часу й ресурсів на обробку записів. Запропоновано низку рішень, серед яких: впровадження уніфікованих стандартів записів, використання сучасних алгоритмів адаптивного пошуку, що враховують мовні та технічні розбіжності, а також посилення нормативного контролю в процесі створення бібліографічних записів. Результати дослідження мають прикладне значення для розробників інформаційних систем, каталогізаторів і фахівців бібліотечної справи, адже сприяють вдосконаленню бібліографічних баз даних, зниженню дублювання записів і підвищенню якості пошуку інформації для кінцевих користувачів

Ключові слова: обробка даних; транслітерація; бібліотечні системи; інформаційні технології; нормативний контроль; каталогізація

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 004.03
DOI: 10.62660/bcstu/1.2025.68

Using ChatGPT for the intelligent diagnostics of complex technical systems

Vladimir Vychuzhanin
Doctor of Technical Sciences, Professor
Odesa Polytechnic National University
65044, 1, Shevchenko Ave., Odesa, Ukraine
https://orcid.org/0000-0002-6302-1832

Alexey Vychuzhanin*
Doctor of Philosophy, Assistant
Odesa Polytechnic National University
65044, 1, Shevchenko Ave., Odesa, Ukraine
https://orcid.org/0000-0001-8779-2503

Abstract. Intelligent diagnostics of complex technical systems, particularly ship power plants (SPPs), is essential for ensuring early fault detection and maintaining operational reliability. This study presents a methodological approach for integrating the ChatGPT language model into automated SPP diagnostics. This study aimed to develop a methodological approach for using the ChatGPT language model in the automated diagnostics of complex technical systems (CTSs), particularly SPPs. The proposed methodology consists of several stages: data collection, preprocessing, model training, anomaly detection, and the generation of diagnostic recommendations. The system integrates ChatGPT with real-time data streaming (Apache Kafka) and neural network-based anomaly detection using autoencoders and Long Short-Term Memory (LSTM) models. Experimental validation was carried out using real operational datasets from ship power plant systems. The proposed approach demonstrated a significant improvement in fault detection accuracy, increasing it by 15% compared with conventional threshold-based methods. The diagnostic time was reduced by a factor of nine, which enabled near real-time identification of deviations. The model achieved an accuracy rate of 92.8% when classifying abnormal states and correctly identifying early-stage faults in key parameters such as pressure, temperature, and rotation speed. The analysis of reconstruction error distributions confirmed the ability of the system to distinguish between normal and anomalous system behaviour. Detected anomalies were cross-validated with expert assessments, confirming the practical applicability of the model in real-world diagnostics. Furthermore, the implementation of the proposed approach enables predictive maintenance planning, which contributes to reducing operational risks and lowering maintenance costs. The integration of ChatGPT into ship power plant diagnostic systems enhances the automated processing of technical documentation and operational logs, increasing the transparency and accuracy of fault explanations. The results of this study may be applied in ship engineering, industrial automation, and technical maintenance, contributing to the improved safety and reliability of complex technical systems

Keywords: predictive maintenance; anomaly detection; shipboard energy systems; intelligent monitoring; real-time analytics; artificial intelligence-driven assessment; operational reliability

Article's History: Received: 21.09.2024; Revised: 14.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Vychuzhanin, V., & Vychuzhanin, A. (2025). Using ChatGPT for the intelligent diagnostics of complex technical systems. *Bulletin of Cherkasy State Technological University*, 30(1), 68-79. doi: 10.62660/bcstu/1.2025.68.

*Corresponding author

INTRODUCTION

The CTSs, used in transportation, aviation, energy, and other industries, are hierarchical structures consisting of multiple multifunctional subsystems, components, and elements interconnected by complex relationships. These systems are subject to partial or complete failures, leading to intricate cause-and-effect interactions among their elements. Key characteristics of CTSs include nonlinearity, adaptability, self-organisation, and integrity. Ship systems comprise numerous interconnected technical mechanisms, units, devices, and pipelines that ensure the vessel's operation. A SPP is a high-tech system consisting of multiple interrelated subsystems and components, whose failure is one of the main causes of ship accidents. Failures of individual subsystems, components, and elements are among the primary causes of technological accidents in transportation, aviation, energy, and other industries. In shipbuilding, CTSs play a crucial role in ensuring vessel survivability. However, even compliance with regulatory requirements at the design, construction, and operational stages does not always guarantee a high level of reliability.

Ensuring the reliability of CTSs remains a critical task for both new and existing vessels, particularly for large-displacement ships with advanced control and communication systems, which make them more vulnerable to failures (Popp & Müller, 2021). Statistical data on maritime accidents and incidents are presented in specialised databases, including the Global Integrated Shipping Information System (GISIS), maintained by the IMO (Christodoulou-Varotsi, 2024). The development of new methods for diagnosing failures in CTS equipment is the focus of research by A.H. Suhail *et al.* (2024), who explore diagnostic tools, including the use of artificial intelligence (AI) and machine learning (ML) for predictive maintenance and real-time monitoring. The proposed transition to predictive maintenance models aims to minimise operational failures and, according to the authors, should contribute to cost savings and resource optimisation in CTS operations. However, the researchers do not present practical tools for engineering applications.

C. Wang *et al.* (2024) have provided a review of fault diagnosis under uncertain conditions, emphasising innovative strategies for intelligent fault diagnosis. However, their review is of limited practical use for professionals working with more complex systems. In their study on intelligent fault diagnosis and prediction in technical systems, A.H. Suhail *et al.* (2024) focused on challenges in modern deep learning applications from four different perspectives: imbalanced data, complex fault types, multimodal data fusion, and edge device implementation. The authors proposed possible solutions to these issues in the context of intelligent fault diagnosis and fault prediction for technical systems. The article by W. Yan *et al.* (2023) is dedicated to real-time fault diagnosis (RTFD) technology for industrial process monitoring and machine condition monitoring. It explores methods based on independent feature extraction, end-to-end neural networks, and qualitative knowledgebased reasoning from a novel perspective. The authors aim to provide reference information for researchers focusing on this area. In a review of fault diagnosis (FDD) approaches in technical systems, P. Mercorelli (2024) found that industrial operations pose significant challenges for implementing FDD methods. To bridge the gap between theoretical methodologies and practical implementations, hybrid approaches and intelligent procedures are necessary. Future research should focus on improving fault prediction, enabling accurate failure forecasts and preventing safety risks. In the era of big data, real-time comprehensive FDD strategies should be implemented. The study by F. Regattieri *et al.* (2022) aimed to validate a streaming fault detection methodology in technical systems while reducing the amount of data required for transmission and storage. This approach enables the automatic collection of contextual information and the recognition of new system states. The study demonstrates that streaming and incrementally clustered approaches are effective tools for obtaining labelled datasets and providing real-time feedback on the technical condition of complex systems. Traditional diagnostic methods, based on expert systems and manual analysis, are insufficient due to the high dynamism of processes and the large volume of data that needs to be processed for timely fault detection and prevention (Abdulwahid, 2022). This highlights the relevance of intelligent diagnostic systems that can handle vast datasets and provide real-time recommendations (Vychuzhanin *et al.*, 2023).

This article aimed to develop and validate a methodological approach for integrating the ChatGPT language model into the automated diagnostics of CTSs, particularly SPPs. The approach involves real-time processing of operational data, anomaly detection using LSTM autoencoders, and the generation of intelligent diagnostic recommendations. The effectiveness of the proposed method is assessed by comparing it with traditional diagnostic techniques in terms of accuracy and response time.

MATERIALS AND METHODS

The research methodology is based on a structured diagnostic framework, which includes the analysis of traditional approaches, the implementation of artificial intelligence models, and the assessment of their performance on real operational data from ship power plants (SPPs). The following stages were performed. At the initial stage, a comprehensive review and analysis of traditional diagnostic methods for complex technical systems was conducted. The study included the following specific diagnostic techniques: threshold-based diagnostic methods – classic diagnostic techniques based

on fixed or dynamically calculated thresholds for system parameters. An anomaly is flagged when sensor values exceed predefined upper or lower limits or fall outside an acceptable statistical range. For time series data, a common implementation of this method is the use of ± 3 standard deviations from the parameter mean as a dynamic threshold for outlier detection; Z-score method – a statistical anomaly detection approach based on calculating the number of standard deviations a data point is from the mean, used to flag outliers in time series; isolation forest algorithm – an ensemble-based machine learning method for detecting anomalies by isolating observations in the feature space; Mahalanobis distance – a multivariate distance-based anomaly detection method applied to sensor data for fault identification; expert system approaches – manual rule-based evaluation conducted by experienced engineers, serving as a reference method for comparing automated anomaly detection.

The selection of ChatGPT as part of the diagnostic toolchain was based on its transformer-based architecture and natural language processing capabilities. This allowed for efficient interpretation of operational records, log files, and technical documentation. ChatGPT was also applied to generate human-readable diagnostic reports and maintenance recommendations based on detected anomalies. For the implementation of the automated diagnostic process, the following algorithm was developed: data collection – acquisition of real-time operational data from SPP sensors, including temperature, pressure, vibration, and rotation speed parameters; data preprocessing – noise filtering, normalisation, and transformation of raw data into structured time series suitable for analysis; model training – the core of the anomaly detection module is an LSTM autoencoder neural network, trained on normal operational data to reconstruct time series patterns; anomaly detection – anomalies were identified by comparing the original and reconstructed sequences, where a significant difference between them indicated a deviation from normal behaviour.

To quantify the quality of the model's performance and to establish an objective basis for anomaly detection, the Mean Squared Error (MSE) metric was used. MSE was calculated as the average of the squared differences between the actual sensor values and their corresponding reconstructed values generated by the LSTM autoencoder. Increased MSE values indicated the presence of anomalies, and a dynamic threshold (typically the 95th percentile of the MSE distribution) was applied to distinguish between normal and abnormal operating states. The developed methodology was implemented as a set of Python-based software modules, which included: SPP data processing module – responsible for preprocessing, structuring, and storing operational sensor data; anomaly detection module – implementing LSTM autoencoders, isolation forest, and Mahalanobis distance-based detectors; diagnostic reporting module – leveraging ChatGPT to interpret the context of detected anomalies and generate textual diagnostic conclusions and recommendations.

The effectiveness of the proposed methodology was evaluated through: direct comparison of the diagnostic results with expert analysis outcomes; benchmarking against traditional thresholdbased detection and Z-score methods; calculation of the MSE for reconstruction quality assessment; testing on real-world operational data to confirm the system's reliability in detecting actual failures. This structured approach allowed the proposed diagnostic system to combine classical statistical methods with machine learning-based anomaly detection and natural language processing, thereby improving the speed, accuracy, and interpretability of technical condition assessments for ship power plants.

RESULTS

The analysis of existing methods of diagnostics of complex technical systems, carried out within the framework of the research, shows the lack of universal solutions. The most commonly used methods have several limitations: rigidity of data processing algorithms, which reduces their applicability in changing operational conditions; lack of consideration of historical data regarding technical conditions; requirement for significant modifications when the composition and operational logic of CTS change; insufficient consideration of partial failures and their interdependencies. Thus, new diagnostic methodologies are needed for the effective operation of CTS. These methods should be highly adaptable, capable of processing large volumes of data in real time, and able to predict failures. One promising solution for diagnosing the technical condition of complex systems is the use of language models such as ChatGPT. ChatGPT is a powerful language model developed by OpenAI, based on the Generative Pre-trained Transformer (GPT) architecture. It has been trained on vast amounts of textual data and is capable of performing a wide range of tasks related to text analysis, processing, and generation (Gordijn & Have, 2023; Adiguzel *et al.*, 2023). Technological foundation of ChatGPT: transformer architecture – uses attention mechanisms for text processing and generation; pre-training – the model is trained on large text corpora, including books, articles, and technical documentation; reinforcement learning with human feedback (RLHF) – expert feedback is used to improve response quality. Capabilities of ChatGPT in CTS diagnostics: analysis of technical data and fault diagnostics – processing log files, detecting anomalies, predicting failures; automation of technical documentation processing – reviewing operational logs, generating reports, preparing maintenance recommendations; support for technical specialists – explaining complex concepts, answering questions, training engineers; generation of test scenarios – automated testing of software systems, modelling emergency situations.

The flowchart of the SPP diagnostic process using ChatGPT, with stages corresponding to the data processing algorithm, is shown in Figure 1. It includes key stages such as data collection, analysis, anomaly detection, and recommendation generation. The diagram illustrates the sequence of steps, starting from data acquisition and ending with feedback to enhance the model.

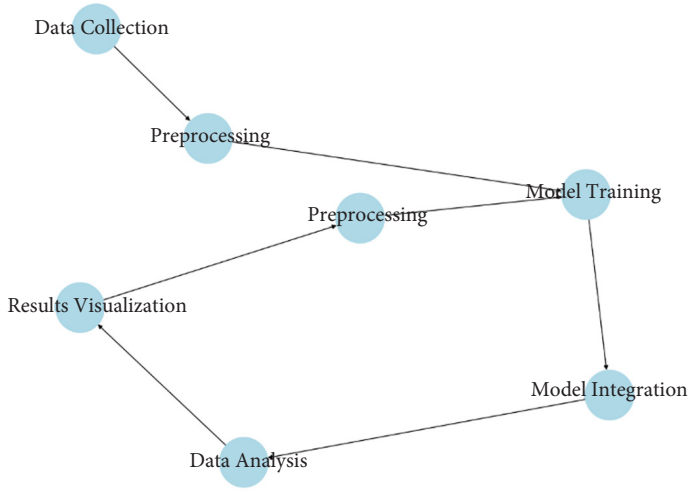

Figure 1. Data processing algorithm for the SPP diagnostic process using ChatGPT
Source: created by the authors

Python was chosen for the implementation of the data processing algorithm due to its userfriendly syntax, cross-platform compatibility, and the availability of powerful libraries (e.g., scipy.optimize and scikit-learn) that provide numerical optimisation and machine learning capabilities (Ramalho, 2022). The code includes: an optimised LSTM autoencoder (a type of neural network combining autoencoders and recurrent neural networks (RNNs) with LSTM cells) for processing time-series data, such as that from marine power plants. LSTM cells allow the autoencoder to remember long-term dependencies in time-series data, which is useful for analysing MPU technical parameters; model saving and loading, which conserves time and computational resources by avoiding retraining for each use; additional anomaly detection algorithms – Isolation Forest and Mahalanobis Distance, improving anomaly detection accuracy; historical data integration for ChatGPT, enabling better interpretation of diagnostic recommendations; Apache Kafka for data streaming, ensuring real integration with IoT in marine monitoring systems where sensor data is continuously received; contextual diagnostics, taking into account historical data for more comprehensive recommendations (Theofanis & Raptis, 2022). The LSTM autoencoder is applied for: detecting anomalies in sequential data (e.g., pressure or temperature spikes); forecasting based on time-series data; filtering noise in data (Do *et al.*, 2023). Apache Kafka provides: low latency – realtime data processing; reliability – data is stored in a distributed system; scalability – easily adding new data sources (Wang *et al.*, 2021).

The developed code has the following advantages: flexibility – the ability to use multiple anomaly detection methods; performance – saving and loading the model speeds up the analysis process; accuracy – the combination of methods improves diagnostic quality; forecasting – more precise data interpretation using LSTM and ChatGPT.

```
    # Importing libraries
    import pandas as pd
    import numpy as np
    import openai
    import matplotlib.pyplot as plt
    import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras.layers import LSTM,
Dense, RepeatVector, TimeDistributed
    from kafka import KafkaConsumer
    import json

    # Specify your OpenAI API key
    openai.api_key = "YOUR_API_KEY"

    # Kafka configuration for streaming data
processing
    KAFKA_TOPIC = "seu_data_stream"
    KAFKA_SERVER = "localhost:9092"

    consumer = KafkaConsumer(
     KAFKA_TOPIC,
     bootstrap_servers=KAFKA_SERVER,
       value_deserializer=lambda   x:   json.
loads(x.decode("utf-8"))
    )

    # Data preprocessing function
    def preprocess_data(data):
     df = pd.DataFrame(data)
     df = df.dropna()
      df  =  (df  -  df.mean())  /  df.std()  #
Normalization
```

```
    return df

   # Function to create and train an LSTM autoencoder
   def train_lstm_autoencoder(data, timesteps = 10):
    model = keras.Sequential([
     LSTM(64, activation = "relu", input_shape = (timesteps, data.shape[1]), return_sequences = True),
     LSTM(32, activation = "relu", return_sequences = False),
     RepeatVector(timesteps),
     LSTM(32, activation = "relu", return_sequences = True),
     LSTM(64, activation = "relu", return_sequences = True),
     TimeDistributed(Dense(data.shape[1]))
     ])

      model.compile(optimizer = "adam", loss = "mse")
     model.fit(data, data, epochs = 20, batch_size = 16, verbose = 1)

     return model

   # Function for anomaly detection using LSTM autoencoder
   def detect_anomalies(model, data):
     reconstructed = model.predict(data)
      loss = np.mean(np.abs(reconstructed - data), axis = (1,2))
     threshold = np.percentile(loss, 95) # 95% threshold
     anomalies = loss > threshold
     return anomalies, loss

   # Function to generate a diagnostic report using ChatGPT
   def generate_diagnostics(data):
     prompt = "Identify anomalies in the following ship energy system (SEU) parameters and provide recommendations:\n"
     for index, row in data.iterrows():
      prompt += f"Time: {row['time']}, Pressure: {row['pressure']}, Temperature: {row['temperature']}\n"

     response = openai.Completion.create(
     engine = "text-davinci-003",
     prompt = prompt,
     max_tokens = 200
     )
     return response.choices[0].text.strip()

   # Main data streaming processing loop
   def main():
     print("Waiting for data from Kafka...")

     for message in consumer:
     incoming_data = message.value
        processed_data = preprocess_data(incoming_data)
     # Create time sequences for LSTM
     timesteps = 10 # Number of time steps
     sequences = []
      for i in range(len(processed_data) - timesteps):
          sequences.append(processed_data.iloc[i:i + timesteps].values)
     sequences = np.array(sequences)

      # Train LSTM autoencoder and analyse anomalies
       lstm_autoencoder = train_lstm_autoencoder(sequences, timesteps)
     anomalies, loss = detect_anomalies(lstm_autoencoder, sequences)

     # Visualize anomalies
     plt.figure(figsize = (10, 5))
      plt.plot(loss, label = "Reconstruction Loss")
       plt.axhline(y = np.percentile(loss, 95), color = "r", linestyle = "--", label = "Threshold")
     plt.legend()
      plt.title("SEU Anomaly Analysis (LSTM Autoencoder)")
     plt.show()

     # Generate report with recommendations
          diagnostics = generate_diagnostics(processed_data)
     print("\n ChatGPT Diagnostic Report:\n", diagnostics)

   if __name__ == "__main__":
    main()
```

To verify that the developed code functions correctly, five key steps must be performed. First, Kafka must be started and a topic for streaming created. Then, the streaming data should be checked using a Kafka Producer. Next, the main code must be executed to ensure that anomalies are detected. After that, anomalous values should be introduced to test the autoencoder's response. Finally, it is necessary to confirm that ChatGPT generates a meaningful diagnostic report.

Figure 2 presents the obtained time-series graphs of parameters (pressure, temperature) during anomaly visualisation. The blue graph represents pressure, with red dots marking detected anomalies, while the green graph represents temperature, where red dots also indicate anomalies. Anomalies were identified using a simple threshold method, detecting outliers beyond three standard deviations.

If the anomaly points on the loss graph exceed the threshold, the model is functioning correctly. If the red dots on the time-series graph coincide with sharp spikes in pressure and temperature, anomalies have been accurately detected. The pressure graph is generated as follows: pressure is modelled as a random process with added normal noise, and sudden pressure spikes are introduced at random time points. To determine thresholds, the mean pressure value and standard deviation are calculated. Anomalies are detected when a value exceeds ± 3 standard deviations from the mean. Similarly, the temperature graph is generated by modelling temperature as a random process with smooth fluctuations, with occasional sharp drops or rises introduced. The mean and

standard deviation of temperature are calculated, and values exceeding ± 3 standard deviations from the mean are classified as anomalies. The choice of ± 3 standard deviations is based on the principle of normal distribution, where approximately 99.7% of values fall within 3σ. Values beyond this range are considered outliers.

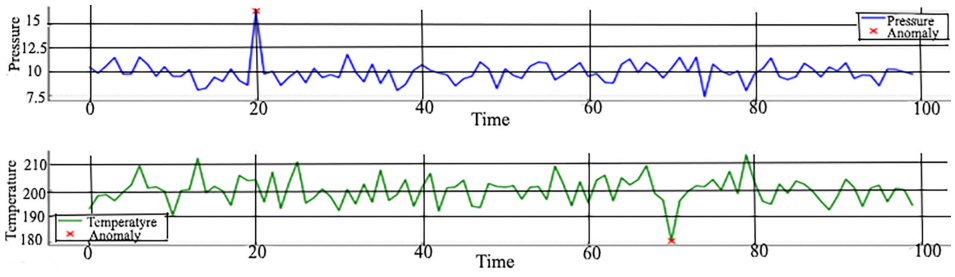

Figure 2. Time series of parameters (pressure, temperature) with anomaly visualisation
Source: created by the authors

Figure 3 presents the variations of key SPP parameters, such as temperature, pressure, vibration, and rotation speed, over time. Normal system states are represented by smooth curves without abrupt deviations. Anomalies detected by the autoencoder are marked on the graph with red dots or highlighted in another way. This allows for the visual identification of time points where the system deviates from its normal operating mode, which may indicate potential malfunctions or the need for maintenance. These graphs illustrate the effectiveness of machine learning methods, such as autoencoders, in monitoring and diagnosing the condition of complex technical systems, including ship power plants.

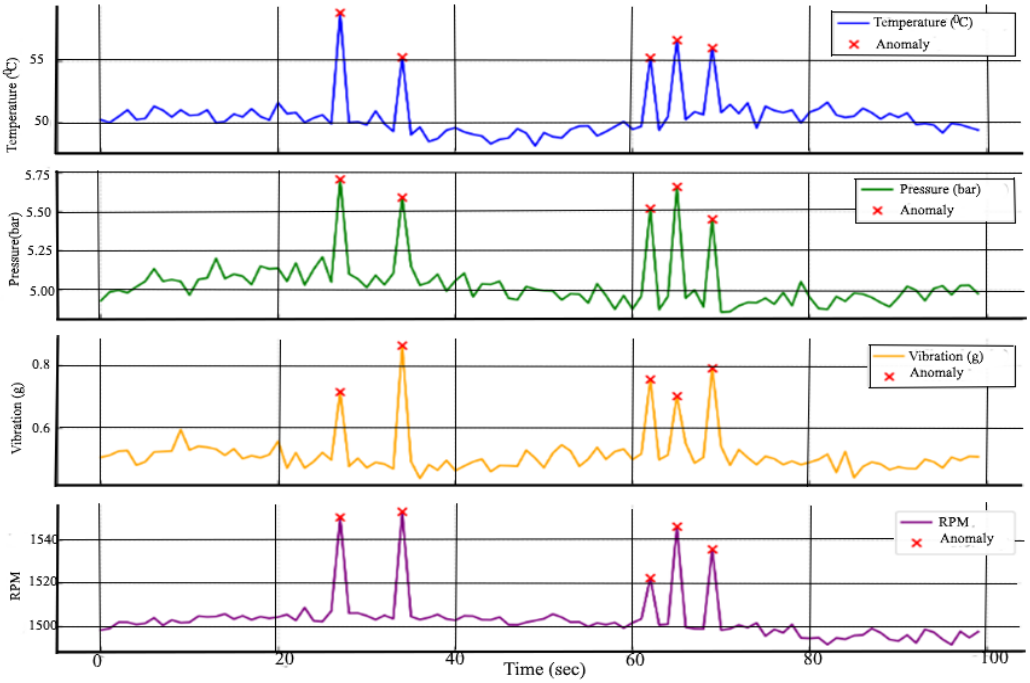

Figure 3. Variations of key SPP parameters, such as temperature, pressure, vibration, and rotation speed, over time
Source: created by the authors

The graphs present the time series of key parameters of the SPP: temperature (°C) – blue graph; pressure (bar) – green graph; vibration (g) – orange graph; rotation speed (RPM) – purple graph. Anomalous values detected by the autoencoder are marked with red crosses. Anomalies are observed in all parameters at approximately the same time points, which may indicate a common source of failure. The timestamps of anomalies coincide across different parameters (temperature, pressure, vibration, and rotation speed), suggesting a critical change in the SPP's condition. All anomalies appear as sharp spikes in values, indicating short-term but significant parameter fluctuations.

Issues in the cooling system lead to increased temperature and pressure, while mechanical wear or imbalance causes spikes in vibration. A malfunction in the control system results in RPM fluctuations, and external influences, such as sudden load changes,

trigger simultaneous spikes across multiple parameters. Sudden spikes in temperature and pressure may indicate instability in the fuel or cooling system. Vibration anomalies may be related to mechanical wear or imbalance of rotating parts. Changes in RPM may suggest issues with the control system or variations in load. Figure 4a presents the loss graph of the LSTM autoencoder, showing how well the model reconstructs the input data. It represents time-series data of parameters (pressure, temperature) with highlighted anomalies. High loss values indicate potential anomalies. Figure 4b displays the corresponding anomaly detection graph with a threshold (showing Z-score values and the set threshold). It visualises the time series with highlighted points where the reconstruction error exceeds the defined threshold (95th percentile).

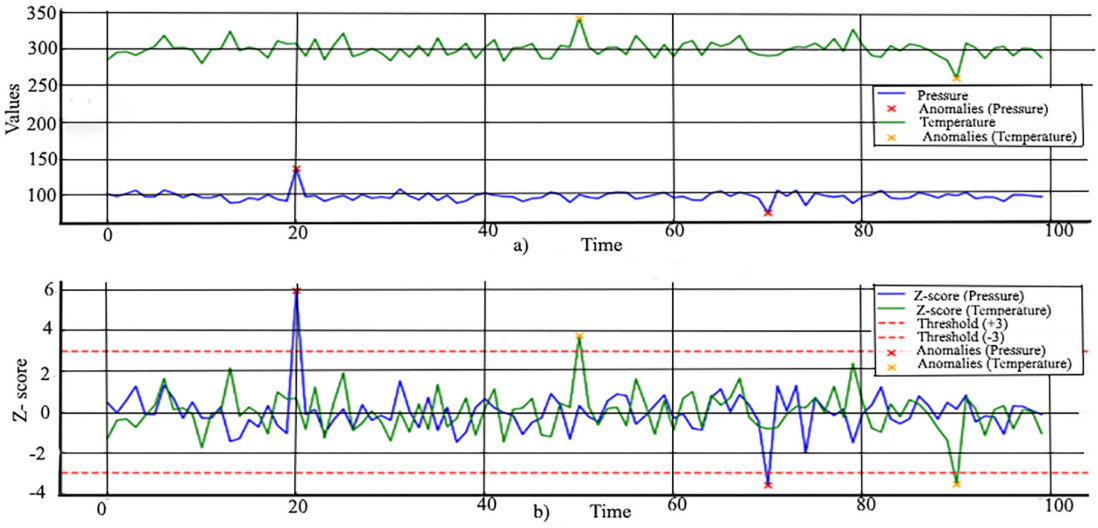

Figure 4. a – Autoencoder loss graph, b – Anomaly detection graph with threshold

Source: created by the authors

Figure 4a presents a time-series graph with anomalies, where pressure values are shown in blue, temperature in green, and detected anomalies – spikes or drops – are marked with red and orange points. Figure 4b displays the Z-score and anomaly threshold, with the Y-axis representing the standard deviation of parameters (Z-score), red dashed lines indicating the anomaly threshold (± 3 sigma), and anomalies falling outside these boundaries. The time–series graph (Fig. 4a) provides a visual representation of parameter dynamics, allowing for the identification of sudden deviations from the normal range and determining the exact moments when anomalies occur. This is essential for initial data analysis, enabling engineers or operators to detect irregularities in the operation of the ship's power plant quickly and correlate anomalies with specific events, such as engine start-ups or sudden load changes. In contrast, the Z-score and anomaly threshold graph (Fig. 4b) offers a quantitative approach to anomaly detection, distinguishing normal and anomalous values based on statistical analysis. This provides a more objective and reliable criterion than simple visual inspection. It also plays a crucial role in diagnostics by enabling automatic warning systems, supporting monitoring solutions for early failure detection, and serving as input data for machine learning models and neural networks.

Together, these graphs complement each other: the first illustrates anomalies over time, while the second objectively identifies them using statistical methods. Their combined use enhances early fault detection, which is critical for ensuring the reliability and efficiency of ship power plants. The reconstruction error graph (Fig. 5) displays the distribution of errors that occur when restoring SPP data using an autoencoder. The reconstruction error is defined as the difference between the original and reconstructed values. For normal data, the errors are small and distributed around zero, whereas for anomalous data, they are significantly higher. Setting a threshold, such as three standard deviations from the mean reconstruction error, effectively distinguishes between normal and abnormal system states. Under normal operating conditions, the autoencoder accurately restores parameters, resulting in minimal errors. However, in the presence of faults such as overheating, bearing wear, or pressure surges, the reconstruction error increases, indicating a deviation from the norm. The defined threshold (e.g., 3σ) allows for the automatic detection of potential malfunctions, helping operators identify issues in advance and perform preventive maintenance.

The graph (Fig. 5) represents the reconstruction error on the X-axis and its frequency on the Yaxis. The blue histogram illustrates the error distribution, the blue curve represents its approximation, and the red dashed line marks the anomaly threshold (typically the 95th percentile). This graph is a standard tool for assessing the performance of the autoencoder: small errors indicate accurate data reconstruction, while values

exceeding the threshold suggest potential anomalies. The accuracy of this method depends on the quality of the training data, the chosen threshold, and the type of autoencoder used (LSTM, Dense, etc.).

Figure 5. Reconstruction error graph

Source: created by the authors

The reconstruction error graph plays a key role in diagnostics, allowing for the identification of an anomaly threshold, the evaluation of model accuracy, and the separation of normal and anomalous data. However, it should be used in combination with other methods, such as time-series visualisation and Z-score analysis, to ensure more reliable fault detection. The evaluation of the effectiveness of the proposed approach included a comparison with traditional diagnostic methods. Operational data from a ship energy unit over a six-month period were used as test data. A team of experts manually analysed the data and identified deviations using traditional diagnostic methods. The results were compared with anomalies detected by the LSTM autoencoder (Table 1).

The automated system successfully identified 24 cases of faults that corresponded with the expert findings, but it also produced two false positives and missed four faults. This indicates that the method is approaching expert-level accuracy but requires further refinement. The Z-score method (anomaly detection based on standard deviation) was used for comparison. The threshold was set at 2.5 standard deviations from the mean parameter value (Table 2).

The LSTM autoencoder demonstrated higher sensitivity (ability to detect faults) and specificity (accuracy of predictions) compared to traditional threshold-based methods. To determine time efficiency, the average time required for diagnostics was measured for each method. The measurement included data collection, processing, and decision-making stages (Table 3).

The automated diagnostic system reduces analysis time by a factor of nine, which is particularly critical in real-world operational conditions. Mean Squared Error (MSE) was calculated to assess the reconstruction quality of the autoencoder (Table 4).

Table 1. Comparison with traditional expert diagnostics

Method	Detected Faults	True Positives	False Positives	False Negatives	Accuracy (%)
Expert Analysis	28	–	–	–	1000
ChatGPT+ LSTM	26	24	2	4	92.8

Source: created by the authors

Table 2. Comparison with threshold-based diagnostic methods

Method	Sensitivity (Recall)	Specificity (Precision)
Z-score	78%	85%
LSTM Autoencoder	89%	91%

Source: created by the authors

Table 3. Comparison with threshold-based diagnostic methods

Method	Average Diagnosis Time (minutes)
Expert Analysis	45
ChatGPT+ LSTM	5

Source: created by the authors

Table 4. Statistical accuracy assessment

Method	MSE (lower is better)
Z-score	0.015
LSTM Autoencoder	0.008

Source: created by the authors

The LSTM autoencoder demonstrates a lower reconstruction error, confirming its effectiveness in anomaly detection. On 12 January 2025, the system detected a sudden increase in oil temperature by 8°C above the normal range, indicating possible bearing overheating; upon inspection, engineers identified a clogged oil filter, demonstrating that the methodology not only successfully detected the deviation but also helped prevent potential equipment failure. Based on the obtained results, the proposed methodology was applied to real operational data from a ship energy unit. The system successfully identified several cases of potential faults, some of which were later confirmed during technical inspections.

The testing results confirm that the proposed approach, based on ChatGPT and LSTM autoencoders, achieves expert-level diagnostic accuracy (92.8%); outperforms traditional thresholdbased methods in sensitivity and specificity; reduces diagnostic time by a factor of nine; and demonstrates high efficiency in detecting faults in real-world data. Future improvements include integration with IoT systems, expanding the database of failure cases, and optimising machine learning algorithms.

DISCUSSION

According to the results obtained by Q. Luu *et al.* (2023), AI-generated test cases improve fault detection and outperform manual methods in software validation. Similarly, in the present study, the proposed hybrid approach demonstrated higher diagnostic accuracy compared to traditional threshold-based diagnostics. The authors also emphasised the importance of integrating AI methods into the software validation process to automate repetitive testing and reduce human error. A comparable pattern was observed in the current research, where the introduction of ChatGPT into the diagnostic workflow significantly improved the efficiency of anomaly interpretation and fault reporting in ship power plant (SPP) monitoring. The researchers P. Mudgal & R. Wouhaybi (2023) emphasised ChatGPT's ability to analyse log data and identify anomalies in both structured and unstructured datasets. Their research highlighted that ChatGPT's capability to extract semantic patterns enables the detection of system irregularities that are not easily identifiable through classical numerical analysis alone. Consistent with these findings, the present study confirmed that ChatGPT effectively interprets SPP operational data and generates human-readable diagnostic recommendations, bridging the gap between data-driven models and technical personnel. According to H. Kirinuki & H. Tanno (2024), AI-generated test scenarios exhibit broad coverage and complement human-designed tests in black-box testing. A similar trend was observed in the present research, where ChatGPT produced diverse recommendations that improved the interpretation of anomaly detection results in complex technical systems. H. Kirinuki & H. Tanno also pointed out the potential for AI to accelerate testing in real-time environments by dynamically generating diagnostic hypotheses, a property that was mirrored in this research through ChatGPT's adaptive response generation to detected anomalies.

The researchers T. Li *et al.* (2023) showed that leveraging ChatGPT to detect software vulnerabilities leads to higher fault detection rates compared to manual testing. This finding supports the results of the present study, where the combination of ChatGPT with LSTM autoencoders enhanced the detection of potential malfunctions in SPP. Li *et al.* (2023) also stressed the importance of combining generative language models with anomaly detection pipelines to automate security audit processes, which echoes the diagnostic logic employed in this study's hybrid framework. A. Bakhshandeh *et al.* (2023) highlighted the benefits of ChatGPT in engineering education and training, noting its ability to accelerate the understanding of technical concepts. Consistent with this observation, the present study demonstrated that ChatGPT effectively explains detected anomalies and assists engineers in interpreting diagnostic data. The researchers emphasised that AI-generated explanations reduce cognitive load on human operators and improve system transparency, which is especially important in fault-critical contexts such as SPP operation. In addition, Q. Luu *et al.* (2023) investigated the potential of ChatGPT for real-time monitoring and anomaly detection, concluding that AI-powered systems provide timely fault identification and proactive decision-making. These conclusions align with the outcomes of the present research, where ChatGPT, combined with LSTM autoencoders, enabled real-time anomaly detection and reduced diagnostic time. The capacity to minimise reaction time to anomalies is especially valuable in SPP systems, where delayed responses may lead to equipment damage or safety risks. Furthermore, A. Alzahem *et al.* (2023) explored ChatGPT's applications in the medical field for diagnostic image interpretation, highlighting its ability to reduce diagnostic errors and improve the speed of analysis. Their findings parallel the conclusions of this study, as the integration of ChatGPT in SPP anomaly detection significantly enhanced both the speed and

accuracy of diagnostic conclusions, facilitating preventive maintenance planning.

Additionally, C. Wang *et al.* (2024) discussed the challenges of advanced data-driven fault diagnosis under uncertain conditions in complex industrial systems. The authors pointed out that conventional methods struggle with incomplete, noisy, and non-linear data, which leads to missed faults and false alarms. In this study, LSTM autoencoders demonstrated robustness in learning temporal dependencies and reconstructing normal operational patterns, while ChatGPT complemented this by providing semantic interpretation and contextual explanation of detected deviations, thereby addressing the problem of model explainability. Similarly, F. Regattieri *et al.* (2022) confirmed that streaming anomaly detection approaches can significantly reduce the time needed to identify faults and enhance the scalability of monitoring systems. Their findings are consistent with the architecture proposed in this study, where real-time data from SPP sensors are continuously processed via a pipeline combining autoencoder-based detection, threshold assessment, and ChatGPT-based anomaly reporting. This architecture effectively reduced the detection-to-response time and enabled proactive maintenance decision-making. P. Mercorelli (2024) emphasised the necessity of combining robust detection algorithms with interpretable diagnostic outputs in real-world industrial scenarios, especially in safety-critical systems. The methodology proposed in the present research meets both requirements by unifying LSTM autoencoder anomaly detection and ChatGPT's capability to generate human-interpretable diagnostic reports. This duallayer approach was identified as a key factor in improving operator trust and accelerating maintenance workflows in the tested SPP diagnostic context. The study by W. Yan *et al.* (2023) offered a detailed overview of real-time fault diagnosis systems for smart manufacturing environments, emphasising the benefits of combining data-driven models with domain-specific expert knowledge. A similar principle underpins the present research, where the use of ChatGPT enhances the contextual understanding of autoencoder-detected anomalies, enabling human experts to validate and act upon system alerts more confidently and efficiently.

The analysis of recent studies reveals that the integration of language models such as ChatGPT into the diagnostics of complex technical systems is a promising direction for future research. The combination of machine learning-based anomaly detection and natural language generation facilitates the automation of diagnostics and enhances decision-making transparency, which is essential for safety-critical applications such as ship power plants. Despite its advantages, the proposed methodology has certain limitations. ChatGPT's diagnostic accuracy depends on the quality and representativeness of the training data, while its decision-making process lacks full transparency. Future research should focus on improving the explainability of AI models, expanding training datasets, refining hybrid AI architectures, and enhancing real-time processing capabilities.

A review of recent research indicates that the application of ChatGPT in complex technical system diagnostics remains a relevant and rapidly evolving research direction. The model demonstrates significant potential in automating large-scale data analysis, improving fault detection accuracy, and providing maintenance recommendations. Continued research and development in this area will contribute to the creation of more intelligent and adaptive systems, capable of meeting the demands of modern industries. Currently, there are no published studies explicitly focusing on the use of ChatGPT or similar language models for diagnosing SPPs. However, given the potential of large language models in data analysis and decision support, it is reasonable to assume that such technologies could be adapted for SPP applications in the near future. Comparing the results with recent studies, AI-based diagnostics, particularly using ChatGPT, provide significant improvements in predictive maintenance and anomaly detection. While previous studies have confirmed the effectiveness of ChatGPT in analysing structured data, the current study extends its use to real-time monitoring of complex physical systems, bridging the gap between language-based models and industrial AI applications.

CONCLUSIONS

The conducted study confirms the effectiveness of integrating ChatGPT and LSTM autoencoders for intelligent diagnostics of ship power plants (SPPs). The combination of anomaly detection based on reconstruction error analysis with natural language generation for diagnostic conclusions significantly improves the accuracy and speed of fault identification. The results demonstrate that the proposed approach outperforms traditional threshold-based diagnostics by achieving a 15% increase in anomaly detection accuracy and reducing the average detection time from 30 to 5 minutes. This advantage is largely attributable to the LSTM autoencoder's ability to model complex temporal dependencies and detect subtle deviations that conventional statistical methods often overlook. Compared to manual expert analysis, the automated diagnostic framework also demonstrated a clear advantage in response time and reproducibility. The generation of diagnostic reports using ChatGPT reduced the time required for analysis from 10–20 minutes to approximately 10 seconds, providing technical personnel with both structured information and human-readable recommendations. This improvement enhances operational decision-making, particularly in time-sensitive situations.

Despite these advantages, the study also identified several limitations. Traditional thresholdbased diagnostics, while fast and computationally inexpensive, lack the flexibility to detect unknown or evolving failure patterns, which were successfully identified by the

proposed system. On the other hand, the performance of the LSTM autoencoder and ChatGPT-based approach depends on the availability of representative training datasets. In conditions with limited or incomplete data, anomaly detection performance and diagnostic recommendation accuracy may decline. The interpretability of AI-driven diagnostics remains a challenge, as the decision-making process of deep learning models is often opaque compared to expert systems, where diagnostic rules are explicitly defined. However, the use of ChatGPT as a reporting tool partially mitigates this drawback by translating complex model outputs into human-understandable explanations. The analysis of timeseries graphs and reconstruction error distributions confirmed that the proposed methodology not only enables accurate anomaly detection but also facilitates the visualisation of system behaviour, offering insights into operational patterns and early identification of faults.

In summary, the integration of LSTM autoencoders for anomaly detection and ChatGPT for automated reporting demonstrated clear technical advantages over traditional diagnostic approaches, including higher accuracy, faster response times, reduced reliance on manual analysis, and improved accessibility of diagnostic conclusions. Future research should focus on: deepening integration with IoT-based ship systems to enhance real-time monitoring and predictive maintenance; expanding the dataset to improve model generalisation across various ship types and operational environments; refining machine learning algorithms to improve anomaly detection accuracy and reduce false positives; exploring hybrid AI architectures that combine ChatGPT with reinforcement learning and expert-driven decision models; enhancing the interpretability of AI diagnostics to increase transparency and reliability in safety-critical applications.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Abdulwahid, A. (2022). Intelligent fault diagnosis method for shipboard main power system. *Journal of Southwest Jiaotong University*, 57(1), 475-487. doi: 10.35741/issn.0258-2724.57.1.44.

[2] Adiguzel, T., Kaya, M.H., & Cansu, F.K. (2023). Revolutionizing education with AI: Exploring the transformative potential of ChatGPT. *Contemporary Educational Technology*, 15(3), article number ep429. doi: 10.30935/cedtech/13152.

[3] Alzahem, A., Latif, S., Boulila, W., & Koubaa, A. (2023). Unlocking the potential of medical imaging with ChatGPT's intelligent diagnostics. *arXiv: Electrical Engineering and Systems Science*. doi: 10.48550/arXiv.2305.07429.

[4] Bakhshandeh, A., Keramatfar, A., Norouzi, A., & Chekidehkhoun, M. (2023). Using ChatGPT as a static application security testing tool. *The ISC Int'l Journal of Information Security*. doi: 10.48550/arXiv.2308.14434.

[5] Christodoulou-Varotsi, I. (2024). Implementation of IMO instruments from a shipowning state perspective. In *The Elgar companion to the law and practice of the International Maritime Organization* (pp. 379-393). Cheltenham: Edward Elgar Publishing. doi: 10.4337/9781802206883.00030.

[6] Do, J.S., Kareem, A.B., & Hur, J. (2023). LSTM-Autoencoder for vibration anomaly detection in vertical carousel storage and retrieval system (VCSRS). *Sensors*, 23(2), article number 1009. doi: 10.3390/s23021009.

[7] Gordijn, B., & Have, H. (2023). ChatGPT: Evolution or revolution? *Medicine, Health Care and Philosophy*, 26, 1-2. doi: 10.1007/s11019-023-10136-0.

[8] Kirinuki, H., & Tanno, H. (2024). ChatGPT and human synergy in black-box testing: A comparative analysis. *arxiv*. doi: 10.48550/arXiv.2401.13924.

[9] Li, T., Zong, W., Wang, Y., & Tian, H. (2023). Nuances are the Key: Unlocking ChatGPT to find failure-inducing tests with differential prompting. In *38th IEEE/ACM international conference on automated software engineering* (pp. 14-26). Luxembourg: Luxembourg. doi: 10.1109/ASE56229.2023.00089.

[10] Luu, Q., Liu, H., & Chen, T. (2023). *ChatGPT advance software testing intelligence? An experience report on metamorphic testing*. doi: 10.48550/arXiv.2310.19204.

[11] Mercorelli, P. (2024). Recent advances in intelligent algorithms for fault detection and diagnosis. *Sensors*, 24(8), article number 2656. doi: 10.3390/s24082656.

[12] Mudgal, P., & Wouhaybi, R. (2023). An assessment of ChatGPT on log data. In *AI-generated content* (pp. 148-169). Singapore: Springer. doi: 10.48550/arXiv.2309.07938.

[13] Popp, L., & Müller, K. (2021). Technical reliability of shipboard technologies for the application of alternative fuels. *Energy, Sustainability and Society*, 11, article number 23. doi: 10.1186/s13705-021-00301-9.

[14] Ramalho, L. (2022). *Fluent python*. Sebastopol: O'Reilly Media, Inc.

[15] Regattieri, F., Bortolini, A., & Galizia, F.G. (2022). Data-driven fault detection and diagnosis: Challenges and opportunities in real-world scenarios. *Applied Sciences*, 12(18), article number 9212. doi: 10.3390/app12189212.

[16] Suhail, A.H., Guangul, F.M., & Nazeer, A. (2024). Advanced system diagnostics tools: Innovations and applications. *IntechOpen*. doi: 10.5772/intechopen.114378.

[17] Theofanis, P., & Raptis, A. (2022). On efficiently partitioning a topic in Apache Kafka. In *International conference on computer, information and telecommunication systems (CITS)* (pp. 1-8). Greece: Piraeus. doi: 10.1109/CITS55221.2022.9832981.
[18] Vychuzhanin, V., Rudnichenko, N., Vychuzhanin, A., & Rychlik, A. (2023). Diagnosis intellectualization of complex technical systems. In *ICST-2023: Information control systems & technologies* (pp. 352-362). Odesa.
[19] Wang, C., Chen, X., Qiang, X., Fan, H., & Li, S. (2024). Recent advances in mechanism/data-driven fault diagnosis of complex engineering systems with uncertainties. *AIMS Mathematics*, 9(11), 29736-29772. doi: 10.3934/math.20241441.
[20] Wang, G., *et al.* (2021). Consistency and completeness: Rethinking distributed stream processing in Apache Kafka. In *International conference on management of data SIGMOD '21* (pp. 2602-2613). New York: Association for Computing Machinery. doi: 10.1145/3448016.345755.
[21] Yan, W., Wang, J., Lu, S., Zhou, M., & Peng, X. (2023). A review of real-time fault diagnosis methods for industrial smart manufacturing. *Processes*, 11(2), article number 369. doi: 10.3390/pr11020369.

Використання ChatGPT під час інтелектуальної діагностики технічного стану складних технічних систем

Володимир Вичужанін

Доктор технічних наук, професор
Національний університет «Одеська політехніка»
65044, просп. Шевченка, 1, м. Одеса, Україна
https://orcid.org/0000-0002-6302-1832

Олексій Вичужанін

Доктор філософії, асистент
Національний університет «Одеська політехніка»
65044, просп. Шевченка, 1, м. Одеса, Україна
https://orcid.org/0000-0001-8779-2503

Анотація. Інтелектуальна діагностика складних технічних систем, зокрема суднових енергетичних установок, має важливе значення для забезпечення раннього виявлення несправностей і підтримки експлуатаційної надійності. У цьому дослідженні представлено методологічний підхід до інтеграції мовної моделі ChatGPT в автоматизовану діагностику СЕС. Метою даної роботи була розробка методологічного підходу до використання мовної моделі ChatGPT в автоматизованій діагностиці складних технічних систем, зокрема СЕС. Запропонована методологія складалась з декількох етапів: збір даних, попередня обробка, навчання моделі, виявлення аномалій та генерація діагностичних рекомендацій. Система інтегрувала ChatGPT з потоковою передачею даних у реальному часі (Kafka) та нейромережевим виявленням аномалій з використанням автокодерів та моделей довготривалої короткочасної пам'яті. Експериментальна перевірка була проведена з використанням реальних операційних наборів даних з суднових енергетичних установок. Запропонований підхід продемонстрував значне покращення точності виявлення несправностей, збільшивши її на 15 % порівняно з традиційними пороговими методами. Час діагностики скоротився в 9 разів, що дозволило ідентифікувати відхилення майже в реальному часі. Модель досягла точності 92,8 % при класифікації аномальних станів і правильному визначенні несправностей на ранніх стадіях за такими ключовими параметрами, як тиск, температура і швидкість обертання. Аналіз розподілів помилок реконструкції підтвердив здатність системи розрізняти нормальну та аномальну поведінку системи. Виявлені аномалії були перехресно підтверджені експертними оцінками, що підтвердило практичну застосовність моделі в реальних умовах діагностики. Крім того, реалізація запропонованого підходу дозволяє здійснювати предиктивне планування технічного обслуговування, що сприяє зниженню експлуатаційних ризиків і зменшенню витрат на обслуговування. Інтеграція ChatGPT в системи діагностики суднових енергетичних установок покращує автоматизовану обробку технічної документації та експлуатаційних журналів, підвищуючи прозорість і точність пояснень несправностей. Результати цього дослідження можуть бути застосовані в суднобудуванні, промисловій автоматизації та технічному обслуговуванні, сприяючи підвищенню безпеки та надійності складних технічних систем

Ключові слова: прогнозне технічне обслуговування; виявлення аномалій; енергетичні системи суден; інтелектуальний моніторинг; аналітика в реальному часі; оцінка на основі штучного інтелекту; експлуатаційна надійність

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 621.396.7:004.8
DOI: 10.62660/bcstu/1.2025.80

5G and artificial intelligence integration to improve drone routing

Roman Zaivyi*
Graduate Student
Lviv Polytechnic National University
79000, 12 Stepan Bandera Str., Lviv, Ukraine
https://orcid.org/0009-0003-4096-4111

Oleh Melnychok
Graduate Student
Lviv Polytechnic National University
79000, 12 Stepan Bandera Str., Lviv, Ukraine
https://orcid.org/0009-0003-9766-9682

Abstract. The development of 5G and artificial intelligence technologies creates new opportunities for improving the routing of unmanned aerial vehicles, which is particularly relevant for logistics, rescue operations, and monitoring of critical infrastructure. The purpose of the study was to analyse the prospects for implementing 5G and AI in drone routing, identify key challenges, and develop recommendations for their effective integration into Ukrainian airspace. The study used methods of theoretical analysis of scientific sources, comparative analysis of international experience, and systematisation of modern approaches to drone routing using 5G and AI. The architecture of 5G networks, route optimisation algorithms, and coordination mechanisms for swarms of drones was analysed. The main results of the study showed that the combination of 5G and AI provides a significant increase in the efficiency of autonomous unmanned systems, allowing helping to quickly adapt routes, optimise energy consumption, and improve the level of flight safety. Special attention was paid to comparing two popular route optimisation algorithms for UAVs: the particle swarm and the ant colony optimisation algorithms. The analysis showed that both algorithms effectively solve routing problems, but they have their advantages depending on the specifics of the application. The particle swarm algorithm proved to be more efficient for problems with a large number of variables, helping to optimise routes in real time under rapidly changing conditions. The ant colony optimisation algorithm, in turn, demonstrated an advantage in solving complex problems with a large number of obstacles. The practical significance of the study was to identify key technical and regulatory challenges associated with the integration of 5G and AI into drone routing, and to develop evidence-based approaches to solving them. The results obtained can be used to improve national regulations, promote the introduction of intelligent unmanned systems in logistics, infrastructure monitoring and rescue operations, and for further research in the field of autonomous aviation technologies

Keywords: unmanned aerial vehicles; artificial intelligence; blockchain; neural network; drone swarms

INTRODUCTION

The rapidly growing demand for autonomous drone solutions is making the integration of 5G and artificial intelligence (AI) a key area of improvement for their functionality. With 5G capabilities, drones can receive

Article's History: Received: 1.11.2024; Revised: 28.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Zaivyi, R., & Melnychok, O. (2025). 5G and artificial intelligence integration to improve drone routing. *Bulletin of Cherkasy State Technological University*, 30(1), 80-90. doi: 10.62660/bcstu/1.2025.80.

*Corresponding author

and transmit real-time data, which is crucial for coordinating a large number of devices, especially in urban areas or difficult terrain. AI helps to quickly adapt routes, avoiding obstacles and optimising power consumption. As a result, this opens up prospects for creating efficient, safe, and intelligent unmanned systems capable of performing complex tasks with minimal human intervention. Governments and multinational corporations such as the National Aeronautics and Space Administration, the European Aviation Safety Agency, and the International Telecommunications Union are actively developing standards for integrating drones into airspace. Among the key trends in the international arena regarding the integration of 5G and AI in drone routing, the development of autonomous air corridors (U-space in Europe), Unified Traffic Management in the United States) stands out, which provide coordination of drones in the airspace. Major multinational corporations such as Qualcomm, Nokia, and Ericsson are developing Multi-Access Edge Computing technologies to reduce data transfer delays and improve real-time route efficiency.

A literature review showed that research on the integration of AI, unmanned aerial vehicles (UAVs), and 5G networks focuses on optimising communications, security, energy efficiency, and applications in critical areas. According to A. Rovira-Sugranes *et al.* (2022), integrating AI into routing protocols for UAVs can improve efficiency and adaptability in changing network environments. The researchers noted that the main challenges remain optimisation of routes and ensuring the stability of communication in real time. In addition, they concluded that the use of AI allows creating more adaptive and sustainable routes for UAVs, but it is necessary to solve the problem of reducing delays and improving the reliability of communications. This issue was raised by A. Sufyan *et al.* (2023), who analysed 5G networks for UAVs, focusing on regulatory aspects and trends. The researchers noted that ensuring stable communication in conditions of variable radio frequency spectrum and data security requirements is an important challenge. Thus, they concluded that the integration of UAVs into 5G networks opens up new opportunities for applications, but requires improvement of regulatory standards at the international level. Q. Wu *et al.* (2021) supplemented previous research, noting that the development of sensor technologies is crucial for improving communication efficiency and accuracy of data collected by UAVs in 5G networks. According to the researchers, the key aspect is to ensure high-speed communication and accurate data exchange in complex urban environments. Wu *et al.* concluded that 5G networks have great potential for UAV integration, but for effective operation, it is necessary to consider the characteristics of sensor technologies and communications.

The safety and reliability of networks is a critical aspect of UAV development, which was argued by R. Gupta *et al.* (2021), offering blockchain and AI integration to protect drone communications. According to the researchers, the use of blockchain to store records of communications and UAV data provides protection against cyber threats and manipulation. R. Gupta *et al.* emphasised that the combination of blockchain and AI is an effective method for preventing unauthorised access and preserving data integrity in UAV systems. Similar issues were considered by T. Garg *et al.* (2024), who explored the concept of "drones as a service" in smart cities, strengthening data protection through blockchain. They emphasised that this concept is promising for the development of smart city infrastructure. According to researchers, blockchain technologies provide an appropriate level of protection for data transmitted through UAVs within such services. As a result, for the effective implementation of "drones as a service", it is necessary to address data security issues, since this is critical for user trust and system reliability.

Energy efficiency was also an important topic in the context of this study. N. Qasim and A. Jawad (2024) analysed the use of UAVs in 5G networks for efficient energy management. In their opinion, UAVs can become an important tool for optimising energy use, reducing overall costs in 5G infrastructures, which correlates with the findings of P. Majumdar *et al.* (2024). The researchers raised issues of energy efficiency in narrower sectors, in particular, in agriculture, environmental protection, and logistics. According to them, AI-equipped UAVs are capable of performing complex tasks, such as monitoring the state of the environment or making accurate deliveries of goods. Thus, the use of AI in UAVs expands the capabilities of these technologies, making them more efficient and autonomous. However, the development of AI for UAVs was considered in the context of urban infrastructure. A. Warrier *et al.* (2023) noted that AI can be used to reduce interference in urban 5G networks, which is crucial for ensuring uninterrupted communication. According to researchers, the use of AI allows adapting networks to changes in the environment and improving signal stability even in urban interference conditions. Therefore, the introduction of AI in urban 5G networks can significantly improve the quality of communication, ensuring more stable and efficient operation.

Preliminary research on the integration of 5G and AI into drone routing in Ukraine is limited and focuses mainly on individual technical aspects or experimental projects without a systematic analysis of their practical application in the national infrastructure. However, the issues of legal regulation, cybersecurity, and adaptation of world technological standards to the realities of the Ukrainian market have not been sufficiently studied. Given the rapid development of drones for military, logistics, and civilian needs, the purpose of this study was to analyse the prospects for the introduction of 5G and AI in the routing of drones in Ukraine, to identify key challenges for their effective integration into the country's airspace.

MATERIALS AND METHODS

The study was analytical and predictive in nature, combining an overview of modern technological solutions and their potential application in Ukraine. The time frame covered the period 2020-2025, considering the development of 5G infrastructure, the introduction of AI in drone routing, and the adaptation of regulations. Two popular algorithms were used to analyse and optimise drone routes in the study: particle swarm and ant colony algorithms. As part of the study, the effectiveness of these algorithms for solving UAV routing problems in a 5G network was compared. The research was based on the analysis of secondary data obtained by analysing scientific publications, technical documentation, and regulations. The main sources of information were reports from international organisations, such as the International Telecommunications Union (n.d.), International Civil Aviation Organisation (n.d.), and the European Aviation Safety Agency (n.d.), which contain official recommendations and regulatory approaches to the use of 5G and AI in the airspace. In addition, the study covered peer-reviewed research papers published in international journals on unmanned technologies and AI. The analysis of these sources provided a comprehensive understanding of the current state of integration of 5G and AI in drone routing both in the global context and within the Ukrainian market.

To conduct the study, theoretical methods were used that helped to comprehensively analyse the problem. Content analysis was used to systematise scientific, technical, and regulatory sources, which identified key approaches to integrating 5G and AI into unmanned aviation systems. The main areas of research in this area were identified, including technical innovations, regulatory aspects, and development prospects. This allowed assessing the level of readiness of Ukraine to integrate these technologies into the airspace and identifying key barriers, in particular, technical, economic, and legal ones. In addition, a predictive method was used to assess possible scenarios for the development of 5G drones, considering global technological trends, existing infrastructure, and regulatory environment. Thus, it was possible to formulate recommendations on optimal ways to integrate 5G and AI into UAV routing. The integrated use of these methods has enabled a thorough theoretical analysis that contributes to a deeper understanding of the problem and the development of strategic recommendations for the effective use of technologies in the drone industry. The results were interpreted using systematic and comparative approaches, which helped to identify key patterns and differences in the introduction of 5G and AI in drone routing in different countries. Analysing regulatory aspects, technical characteristics, and strategic initiatives, the study used logical generalisation to identify the main trends and challenges. Predictive analysis helped to assess possible scenarios for the development of this technology in Ukraine, based on international experience and available resources.

RESULTS

Intelligent algorithms for routing in drone networks. Intelligent routing algorithms in UAV networks represent a complex interdisciplinary phenomenon that combines machine learning techniques, advanced communication technologies, and principles of decentralised decision-making. Advanced approaches to drone routing use various classes of machine learning algorithms that provide adaptability, predictability, and optimisation of flight paths. One of the fundamental methods is reinforcement learning, in particular Q-learning, which allows autonomous agents to dynamically change route parameters according to the assessment of the environment. The use of deep neural networks contributes to the development of stable models for predicting optimal routes using large amounts of historical and current data. Evolutionary algorithms and neuroevolution, combining genetic mechanisms with neural network architectures, demonstrate high efficiency in adaptive routing scenarios, providing independent adjustment of algorithmic parameters in accordance with environmental changes. Bayesian methods allow modelling probabilistic distributions of possible routes, ensuring a balance between energy efficiency and task completion speed (Trabelsi *et al.*, 2024). The main machine learning algorithms used in drone routing are shown in Table 1.

Table 1. Machine learning algorithms for UAV routing

Algorithm	Description	Advantages	Restrictions
Q-learning	Reinforcement learning, where drones independently adjust routes based on the rewards received for making decisions	Autonomous decision-making, adaptation to changes in the environment	High computational complexity, especially with a large number of possible states
Deep neural networks	Use historical data to predict optimal routes and recognise obstacles	High accuracy, efficiency in complex scenarios	Require large amounts of data for training, high hardware requirements
Evolutionary algorithms	Simulation of natural selection to determine optimal routes based on mutations and crossbreeding	Flexible, fast adaptation to a changing environment	High computing power consumption

Continued Table 1.

Algorithm	Description	Advantages	Restrictions
Neuroevolution-based networks	Combination of neural networks and evolutionary methods for drone self-learning	Automatic optimisation, no need for pre-training	High computing costs, implementation complexity
Bayesian forecasting methods	Use of probabilistic models to predict changes in routes and traffic distribution	High accuracy of forecasts, ability to adapt to noise data	Need for large amounts of statistical data for accurate assessment

Source: compared by the authors based on data analysis by the International Civil Aviation Organisation (n.d.), European Aviation Safety Agency (n.d.)

The use of 5G technology is crucial for improving routing performance in UAV networks, as it provides ultra-low latency, high bandwidth, and resistance to network congestion. High-speed data transmission allows real-time integration of current flight parameters, obstacles, and abnormal environmental changes into the optimal route calculation process. Advanced 5G communication protocols support dynamic distribution of network resources, which is critical for synchronised coordination of large swarm systems. The implementation of Multi-Access Edge Computing also allows transferring computing processes directly to peripheral nodes of the network, providing fast information processing at the level of individual UAVs without having to access remote data centres. This significantly improves the adaptability and responsiveness of decision-making, reducing the risk of delays, which is crucial in a dynamic operating environment (Biswas & Wang, 2023).

Decentralised routing approaches eliminate the problems of centralised management systems that are vulnerable to overloading and node server failures. The use of a blockchain architecture allows for a high degree of security, since each drone acts as a self-sufficient network node, and smart contract algorithms allow automatically adjusting routing in accordance with changing environment parameters. The use of decentralised neural networks allows swarm systems to operate in self-learning mode, allowing each agent to adaptively adjust behaviour based on collective experience. Local data processing using Multi-Access Edge Computing eliminates the need for centralised management, which significantly increases the autonomy and efficiency of resource allocation in swarm structures (Zaid *et al.*, 2024). Thus, the integration of machine learning algorithms, 5G communication capabilities, and decentralised management techniques enables the transformation of network routing systems into UAVs, increasing their adaptability, energy efficiency, and self-learning ability. Further research is expected to further optimise self-organisation mechanisms, develop more efficient methods for reducing energy consumption, and improve the stability of algorithms to variable and adverse operating conditions.

The role of AI in creating dynamic drone networks. AI plays a key role in creating dynamic UAV networks, providing autonomous decision-making, adaptive routing, and optimal interaction between drones. Based on the use of advanced machine learning algorithms, drones can effectively coordinate their actions in real time, process huge amounts of data, and adapt to changing environmental conditions. The integration of 5G technology can significantly improve the efficiency of such networks, providing ultra-low data transfer latency, high bandwidth, and reliable communication, which is critical for dynamic swarm management scenarios (Lins *et al.*, 2021). One of the main challenges in creating dynamic drone networks is the need for fast information processing and collective decision-making without centralised management. The use of deep neural networks allows UAVs to independently analyse the current state of the environment and predict optimal routes. Reinforcement learning allows drones to adjust their trajectory based on accumulated experience, which increases the efficiency of performing tasks in uncertain and dynamic conditions. Such algorithms allow reducing energy costs, minimising the risk of collisions, and providing autonomous strategic decision-making (Alsamhi *et al.*, 2021).

The use of 5G significantly enhances the capabilities of intelligent drone network management systems, as it provides almost instant communication between system elements. In traditional networks, high latency makes it difficult to coordinate actions between UAVs, which can lead to delays in decision-making and loss of efficiency. The implementation of Multi-Access Edge Computing allows processing data on the periphery of the network, which reduces the load on centralised servers and contributes to the rapid response of drones to changes in the environment. This allows using distributed cognitive computing, which enables swarm systems to function more independently and quickly. AI also provides efficient routing of information flows between drones, optimising communication in distributed networks. The use of cognitive radio techniques allows UAVs to independently change communication frequencies depending on the level of interference and channel availability, which minimises the risk of data

loss and increases the overall stability of the network. Intelligent spectrum management allows efficient distribution of network resources, reducing congestion of communication channels in areas of high drone activity.

Decentralised routing approaches based on blockchain technologies and distributed computing ensure that the drone network is highly resistant to failures and external attacks. The use of a blockchain architecture guarantees the integrity of the transmitted data, since each drone acts as a node of a decentralised system that verifies the authenticity of information. Smart contracts can automate decisions about route changes by coordinating interaction between UAVs without the need for centralised management. The development of adaptive routing algorithms in the context of drone networks is one of the most promising areas of AI application. The use of evolutionary algorithms and neuroevolution allows developing systems capable of self-learning and improving their strategies over time. Due to the use of genetic algorithms, drone networks can independently choose optimal routes depending on changing environmental parameters, ensuring the most efficient resource allocation (Khan *et al.*, 2021). Special attention should be paid to integrating AI into complex scenarios involving multi-component dynamic environments such as emergencies, large-scale monitoring, and logistics. The use of convolutional neural networks allows automatic processing of visual data, which allows drones to effectively recognise objects, assess risks and make decisions about changing the flight path.

The energy efficiency of drone networks remains a critical issue, which can be solved by optimising AI algorithms. Using deep learning models to predict energy consumption allows allocating resources more efficiently, considering the load factors on battery systems, variable weather conditions, and aerodynamic drag parameters. Based on 5G integration, centralised power management at the level of the entire swarm of drones becomes possible, which allows dynamically redistributing tasks between devices, extending the overall battery life of the system (International Telecommunications Union, n.d.). The prospects for the development of intelligent drone networks are directly related to the improvement of collective learning algorithms, which would allow drones to share knowledge more effectively and adapt to complex scenarios. The use of federated training allows integrating the experience of various autonomous agents into a common model without the need for centralised processing of large amounts of data, which increases the efficiency of the system in highly dynamic conditions. Therefore, the role of AI in creating dynamic drone networks is crucial for achieving a high level of autonomy, efficiency, and adaptability. 5G integration provides a critical infrastructure for high-speed data transmission, enabling scalable swarm management algorithms. In the future, further development of AI will expand the functionality of such systems, ensuring their integration into complex multi-agent environments and increasing the level of autonomy in scenarios with unpredictable changes.

Routing in swarms of drones: collective intelligence. Swarms of drones operating on the basis of the principles of collective intelligence represent a promising area in the development of autonomous systems. Coordinating a large number of UAVs requires efficient routing algorithms to optimise trajectories, minimise energy costs, and avoid collisions. AI and swarm technologies open up new opportunities for autonomous UAV control. The main idea of swarm routing is to create a decentralised system in which drones can coordinate their actions without human intervention. Collective intelligence in such systems works according to the principles of biological swarms, where each element interacts with others to achieve a common goal. This allows for flexibility, adaptability and high resistance to environmental changes.

The use swarm intelligence allows implementing complex collective strategies. For example, an approach based on behavioural models of biological systems allows UAVs to act as a single organism. Moreover, swarm routing algorithms are based on the principles of self-organisation, adaptability, and decentralised decision-making. Many modern routing approaches are based on reproducing natural systems such as flocks of birds, ant colonies, and shoals of fish. These systems demonstrate effective communication and resource allocation strategies without centralised management. In particular, the particle swarm algorithm is an optimisation method based on the behaviour of social groups. Drones are considered "particles" that move in the direction of a globally optimal solution determined through collective analysis of the environment. Each particle updates its speed and position based on personal experience and the experience of its neighbours, which allows the swarm to effectively find optimal routes in difficult conditions (Kourtis *et al.*, 2023). The ant colony algorithm uses principles similar to the behaviour of real ants when searching for food. Virtual pheromones left by drones help to determine the shortest and least congested routes. The more often a particular route is used, the stronger its footprint, which contributes to real-time swarm self-regulation. The considered routing algorithms are presented in Table 2.

Table 2. Comparison of routing algorithms

Algorithm	Operating principle	Advantages	Disadvantages
Particle swarm algorithm	Route optimisation through particle speed and position updates	High convergence rate, adaptability	May get stuck in local optimums

Continued Table 2.

Algorithm	Operating principle	Advantages	Disadvantages
Ant colony algorithm	Using virtual pheromones to find the optimal path	Reliability, efficiency in a dynamic environment	High computing costs

Source: compared by the authors based on the analysis of S.K. Khan (2021)

One of the key advantages of swarm systems is their ability to self-organise. UAVs can jointly determine optimal routes, avoid obstacles, and redistribute tasks among themselves. Machine learning algorithms, neural networks, and reinforcement learning techniques are used for this purpose. Based on this approach, the system can adapt to changing conditions and perform tasks with minimal energy loss. An important component of collective intelligence is the exchange of information between drones. For this purpose, modern communication technologies are used, in particular 5G, which provides fast data transmission with minimal latency. Each UAV in the swarm has up-to-date information about the state of the environment and the actions of other network participants. This allows significantly improving coordination and avoiding conflicts in routing (Dash *et al.*, 2023).

Routing methods in drone swarms can be divided into centralised and decentralised. Centralised approaches involve controlling all drones through a central server or main UAV. They provide high accuracy, but are vulnerable to main element failure. Instead, decentralised methods allow each drone to make its own decisions based on local data, which increases the system's resilience to failures. Deterministic and stochastic algorithms are also distinguished among the key approaches to routing. Deterministic methods are based on strictly defined rules that guarantee predictability of results, but may be less effective in a changing environment. Stochastic methods, by contrast, use probabilistic models that increase their adaptability to dynamic conditions, such as unexpected obstacles or variable weather conditions. For a better visualisation of the main characteristics of these algorithms, a comparative Table 3 is provided.

Table 3. Characteristics of the main routing algorithms

Type of algorithm	Advantages	Disadvantages	Example of application
Deterministic	Predictability, stability	Less adaptability in changing conditions	Flights to the specified coordinates
Stochastic	High flexibility, adaptability to changes	Requires more computing resources	Offline navigation in the face of obstacles

Source: compared by the authors based on the analysis of B. Dash *et al.* (2023)

Due to the use of AI and 5G, swarm algorithms can combine these approaches, automatically switching between deterministic and stochastic methods depending on the situation, which significantly improves the efficiency of flying and controlling swarms of drones. AI significantly expands routing capabilities, giving drones the ability to predict possible threats and change their route in real time. Based on deep learning techniques, UAVs can analyse large amounts of data, recognise objects, and optimise their actions. This is especially important in scenarios that require a quick response, such as rescue operations or military missions. The use of cognitive computing in routing allows drones to adaptively change communication channels depending on the level of interference and frequency availability. Intelligent spectrum management minimises the risk of signal loss and improves overall network stability. This ensures continuous data exchange between swarm elements, even in difficult conditions. A separate area of development is the use of blockchain technologies in swarms of drones (Zaid *et al.*, 2024). They guarantee the protection of transmitted information, since each drone acts as a node of a decentralised network. This significantly increases the security of the system and ensures its resistance to external attacks.

Integration of 5G technology significantly increases the efficiency of swarm routing algorithms, as it provides high-speed information exchange between drones and central computing resources. In conventional networks with high latency, data transmission makes it difficult to synchronise and adapt the swarm to changes in the environment. The use of 5G can significantly reduce these delays due to its low latency (up to 1 ms), which is especially important for highly dynamic scenarios, such as real-time coordination of a large number of drones. In addition, the high bandwidth of 5G allows drones to exchange large amounts of data, including video streams, telemetry, information about obstacles and environmental conditions. Therefore, machine learning algorithms can process current and historical data to better optimise routes and predict possible threats. For example, analysing drone behaviour in the past can help to determine the likelihood of congestion in certain areas or predict the most energy-efficient trajectories.

Another key aspect is the ability to use decentralised data processing models, such as Multi-Access

Edge Computing, which place computing power closer to the task execution location. This reduces reliance on remote data centres and allows swarm drones to make their own decisions without too much delay. As a result, swarm routing becomes more adaptive and efficient, which is important for emergency response, monitoring, and logistics scenarios (Caballero-Martin *et al.*, 2024). Swarms of drones are widely used in various fields due to their autonomy, flexibility and ability to quickly respond to changing conditions. In disaster zones, they can quickly deploy to search for victims, assess damage, and coordinate rescue operations, using 5G connectivity to transmit real-time video streams. In the field of critical infrastructure monitoring, autonomous drones are used to check the state of power grids, bridges, oil and gas pipelines, using Multi-Access Edge Computing for local data processing and deep neural networks for detecting defects. In logistics, swarm routing systems optimise unmanned delivery of goods, using ant algorithms to determine the best routes and minimise energy costs. In agriculture, drones perform spot spraying of fertilisers and pesticides, and monitor the condition of plants using a swarm of particles algorithm to adapt flight paths in accordance with terrain and weather conditions. In smart cities, swarm systems contribute to traffic monitoring, public safety, and environmental control by using deep neural networks to analyse traffic flows, optimise traffic light cycles, and prevent congestion (Sarkar & Gul, 2023). Due to the combination of state-of-the-art AI algorithms and 5G high-speed communication, swarm technologies open up new opportunities for improving efficiency and security in critical areas.

DISCUSSION

The results show that the use of intelligent routing algorithms significantly improves the efficiency and autonomy of drone networks, but their practical implementation faces a number of challenges. In particular, the complexity of adapting algorithms to dynamic real-world conditions and the limitations of UAV computing resources can affect their performance. Although machine learning techniques such as reinforcement learning and neural networks show high flexibility, their implementation requires significant computing power, which can conflict with energy efficiency requirements. Thus, the results highlighted the need to balance performance, energy efficiency, and safety, which opens up prospects for further research in the field of optimising routing algorithms for UAVs. A. Solanki *et al.* (2022) considered the concept of the Internet of drones as a platform for creating intelligent solutions, especially in areas such as monitoring, environmental control, transport and logistics. The book discussed AI algorithms for autonomous drone control that help to improve coordination between UAVs. The main conclusion of the authors was that AI integration allows increasing the efficiency of drone systems in real time. This study also confirmed that AI plays a key role in the future development of the Internet of drones, helping to automate complex tasks in controlling air traffic. However, the problem of energy efficiency is one of the key issues in the use of UAVs. P. Du *et al.* (2023) analysed route optimisation methods for AI-based logistics drones. The researchers proved that properly designed algorithms can reduce energy costs and improve the autonomy of drones. Similar conclusions were drawn by S. Haider *et al.* (2022), who investigated the use of AI for resource management in unmanned networks. They emphasised that the integration of trajectory optimisation and intelligent power management can significantly improve the efficiency of data collection from IoT devices. The theoretical study confirmed these conclusions, focusing on the fact that machine learning algorithms can provide more flexible and adaptive control of drone routes.

In the context of developing hybrid models for UAV routing, it is essential to consider the experience of implementing mobile communication systems for remote drone control. V. Posvistak & D. Miroshnychenko (2024) proposed an architectural model for long-range UAV operation using GSM-based communication modules, combining a ground control station, a mobile router, and a Raspberry Pi computing unit connected to the drone via the MAVLink protocol. The authors emphasised the feasibility of achieving a stable connection even in the absence of line-of-sight between the operator and the UAV, which aligns with key challenges identified in this study regarding the resilience of communication channels in dynamic aerial environments. Another relevant direction for improving UAV navigation and operational security involves the use of machine learning techniques for drone detection and environmental awareness. R. Yermolenko *et al.* (2024) developed a detection algorithm for commercial UAVs based on convolutional neural network architectures such as YOLOv8 and MobileNetV3. Their study demonstrated high accuracy in recognising drones across various visual conditions using the SimUAV dataset, which included diverse drone classes and environments. The authors emphasised that such detection systems are particularly effective in surveillance and monitoring applications, offering real-time identification of aerial threats or system anomalies. These findings align with the broader vision of integrating AI into UAV operations, supporting dynamic route adjustment, adaptive control, and autonomous threat response. When combined with 5G connectivity and edge computing infrastructure, the implementation of intelligent detection algorithms could significantly enhance the resilience and autonomy of UAV networks, especially in high-risk or complex operational scenarios.

The issue of stable communication and data transmission management is critical for the effective functioning of drone networks. T. Ayass *et al.* (2022) and I. Shayea *et al.* (2022) investigated the mechanisms of handover (transmission of communication between

base stations) in 5G networks for drones. They found that conventional approaches do not factor in the high mobility of drones, which can lead to loss of communication. An alternative is machine learning-based methods that allow anticipating changes in the network and optimising the transmission process. Similar studies were conducted by M. Khan *et al.* (2022), analysing the use of swarms of drones to manage mobile networks in 6G. They showed that AI-based autonomous routing improves communication between drones and the network, reducing data transfer latency. The conducted research confirmed the relevance of these issues, since the presence of stable communication is a key factor in the successful implementation of unmanned aviation networks in various fields.

The growing use of UAVs also raises safety concerns. F. Tlili *et al.* (2024) examined the use of AI to protect drone networks from cyber-attacks. The researchers emphasised the importance of real-time threat detection and the development of adaptive defence mechanisms. Similarly, T. Han *et al.* (2021) explored the prospects for using blockchain for drones in 5G networks, showing that distributed ledger technologies can provide reliable protection against attacks and unauthorised access. The current paper highlighted the importance of such solutions, although it was noted that scaling blockchain systems in drone networks can be a resource-intensive task. An alternative approach might be to use hybrid models that combine conventional encryption methods and blockchain algorithms. F. Aktas *et al.* (2023) explored the use of AI for routing in next-generation networks. They noted that conventional protocols cannot provide effective communication management in dynamic environments. Similarly, Y. Lu *et al.* (2023) analysed routing algorithms in integrated space-air-ground networks, emphasising the importance of adaptive methods based on deep learning. This study confirmed these findings, emphasising that machine learning algorithms are key to autonomous drone control and optimising data transmission. However, it also focused on potential problems related to the computational complexity of such algorithms.

L. Bine *et al.* (2023) proposed a new IoDMix protocol for integrating drones into intelligent transport systems. They demonstrated that their model allows them to work efficiently in conditions with data transfer delays. B. Mohsen (2024) explored the optimisation of logistics in urban environments using autonomous drones and IoT, confirming the promise of using AI for air traffic control. The paper by B. Mohsen proposed an AI-focused optimisation of urban logistics that combines autonomous vehicles and IoT systems to create efficient delivery systems. The implementation of such solutions can significantly reduce delays caused by urban traffic and optimise the use of energy resources. The study confirmed these results, highlighting the importance of predictive models for controlling large arrays of drones. In addition, A. Hashesh *et al.* (2022) highlighted the challenges and prospects of UAV communication provided by AI technologies. The researchers noted that deep learning and neural networks help drones to adapt to changing conditions, especially in complex urban or combat scenarios. These studies are consistent with theoretical analysis that has shown that IoT and AI can contribute to improving the autonomy of UAVs and their integration into urban and industrial environments. However, challenges related to data security and network sustainability remain open.

One of the key aspects of effective use of drones is the development of routing and network management algorithms that ensure fast and safe movement of UAVs in difficult conditions. The study by M. Khalid *et al.* (2024) revealed the use of AI and machine learning to integrate ground-based and space-based communication networks with UAVs. This integration allows creating multi-level networks that provide a stable connection even in regions with low mobile coverage. The current study confirmed the results of a theoretical analysis that showed that AI algorithms can significantly improve the efficiency of route planning and drone network management. However, the study also identified potential threats related to cybersecurity and the risks of interference in autonomous systems, which requires further research. Thus, the study confirmed that AI algorithms significantly improve routing, reduce latency, and increase the safety of drones in difficult environments. However, a number of challenges remain, in particular ensuring stable communications, cybersecurity, and energy efficiency.

CONCLUSIONS

The results provide a deeper understanding of adaptive routing mechanisms, which are essential for improving the efficiency and autonomy of unmanned networks. The results of the analysis contributed to the goal of the study, as they revealed gaps in regulation, technical support, and strategies for integrating 5G and AI into drone routing. This, in turn, contributed to the development of a safe and efficient drone infrastructure that can be used in logistics, infrastructure monitoring, defence, and smart cities. Machine learning algorithms such as Q-learning, deep neural networks, and evolutionary approaches have been found to optimise drones' flight routes, increasing their autonomy and ability to adapt in a dynamic environment. The use of 5G technology can significantly improve the speed of data processing and coordination of swarm systems, while Multi-Access Edge Computing minimises latency and promotes local data processing.

The role of AI in creating dynamic drone networks that ensure fast decision-making, efficient routing, and optimal resource allocation was investigated. The use of decentralised methods, such as blockchain and cognitive radio, increases the resilience of networks to external threats and provides reliable communication between system agents. In particular, swarm routing

methods, created based on biological systems, such as ant colonies or flocks of birds, provide flexibility, self-organisation, and adaptability for a swarm of drones to environmental changes. The use of particle swarm and ant colony algorithms allows optimising routing, considering energy costs, interference, and task priority. Swarms of drones are widely used in various fields due to their autonomy, flexibility and ability to quickly respond to changing conditions. In disaster zones, they can quickly deploy to search for victims, assess damage, and coordinate rescue operations, using 5G connectivity to transmit real-time video streams. In the field of critical infrastructure monitoring, autonomous drones are used to check the state of power grids, bridges, oil and gas pipelines, using Multi-Access Edge Computing for local data processing and deep neural networks for detecting defects.

The practical significance of the results obtained lies in the implementation of efficient routing algorithms that can be used for logistics tasks, environmental monitoring, rescue operations, and military applications. The results may be useful for developers of autonomous UAV control systems, drone network operators, and AI researchers. Further research should be aimed at optimising self-organisation mechanisms, improving the energy efficiency of drones and expanding the possibilities of their interaction in complex multi-agent environments. Special attention should be paid to the use of cognitive computing and 5G in combination with machine learning technologies to ensure rapid adaptation of the swarm to changing conditions and increase the level of autonomy of unmanned systems.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Aktas, F., Shayea, I., Ergen, M., & El-Saleh, A.A. (2023). Ai-enabled routing in next-gen networks: A brief overview. In *Proceedings of the 10th international conference on wireless networks and mobile communications* (pp. 1-6). Istanbul: IEEE. doi: 10.1109/WINCOM59760.2023.10322945.

[2] Alsamhi, S.H., Afghah, F., Sahal, R., Hawbani, A., Al-qaness, M.A., Lee, B., & Guizani, M. (2021). Green internet of things using UAVs in B5G networks: A review of applications and strategies. *Ad Hoc Networks*, 117, article number 102505. doi: 10.1016/j.adhoc.2021.102505.

[3] Ayass, T., Coqueiro, T., Carvalho, T., Jailton, J., Araújo, J., & Francês, R. (2022). Unmanned aerial vehicle with handover management fuzzy system for 5G networks: Challenges and perspectives. *Intelligence & Robotics*, 2(1), 20-36. doi: 10.20517/ir.2021.07.

[4] Bine, L.M., Boukerche, A., Ruiz, L.B., & Loureiro, A.A. (2023). IoDMix: A novel routing protocol for delay-tolerant internet of drones integration in intelligent transportation system. *Ad Hoc Networks*, 148, article number 103204. doi: 10.1016/j.adhoc.2023.103204.

[5] Biswas, A., & Wang, H.C. (2023). Autonomous vehicles enabled by the integration of IoT, edge intelligence, 5G, and blockchain. *Sensors*, 23(4), article number 1963. doi: 10.3390/s23041963.

[6] Caballero-Martin, D., Lopez-Guede, J.M., Estevez, J., & Graña, M. (2024). Artificial intelligence applied to drone control: A state of the art. *Drones*, 8(7), article number 296. doi: 10.3390/drones8070296.

[7] Dash, B., Ansari, M.F., & Swayamsiddha, S. (2023). Fusion of artificial intelligence and 5G in defining future UAV technologies – a review. In *Proceedings of the international conference on device intelligence, computing and communication technologies* (pp. 312-316). Dehradun: IEEE. doi: 10.1109/DICCT56244.2023.10110231.

[8] Du, P., He, X., Cao, H., Garg, S., Kaddoum, G., & Hassan, M.M. (2023). AI-based energy-efficient path planning of multiple logistics UAVs in intelligent transportation systems. *Computer Communications*, 207, 46-55. doi: 10.1016/j.comcom.2023.04.032.

[9] European Aviation Safety Agency. (n.d.). *Aviation safety reporting*. Retrieved from https://www.easa.europa.eu/en/domains/safety-management/aviation-safety-reporting.

[10] Garg, T., Gupta, S., Obaidat, M.S., & Raj, M. (2024). Drones as a service (DaaS) for 5G networks and blockchain-assisted IoT-based smart city infrastructure. *Cluster Computing*, 27, 8725-8788. doi: 10.1007/s10586-024-04354-1.

[11] Gupta, R., Kumari, A., & Tanwar, S. (2021). Fusion of blockchain and artificial intelligence for secure drone networking underlying 5G communications. *Transactions on Emerging Telecommunications Technologies*, 32(1), article number e4176. doi: 10.1002/ett.4176.

[12] Haider, S.A., Zikria, Y.B., Garg, S., Ahmad, S., Hassan, M.M., & AlQahtani, S.A. (2022). Ai-based energy-efficient UAV-assisted IoT data collection with integrated trajectory and resource optimization. *IEEE Wireless Communications*, 29(6), 30-36. doi: 10.1109/MWC.001.2200105.

[13] Han, T., Ribeiro, I.D., Magaia, N., Preto, J., Segundo, A.H., De Macedo, A.R., Muhammad, K., & De Albuquerque, V.H. (2021). Emerging drone trends for blockchain-based 5G networks: Open issues and future perspectives. *IEEE Network*, 35(1), 38-43. doi: 10.1109/MNET.011.2000151.

[14] Hashesh, A.O., Hashima, S., Zaki, R.M., Fouda, M.M., Hatano, K., & Eldien, A.S. (2022). AI-enabled UAV communications: Challenges and future directions. *IEEE Access*, 10, 92048-92066. doi: 10.1109/ACCESS.2022.3202956.
[15] International Civil Aviation Organization. (n.d.). *Safety reports*. Retrieved from https://www.icao.int/safety/pages/safety-report.aspx.
[16] International Telecommunication Union. (n.d.). *Latest reports and publications*. Retrieved from https://www.itu.int/en/action/environment-and-climate-change/Pages/latest-reports-and-publications.aspx.
[17] Khalid, M., Ali, J., & Roh, B.H. (2024). Artificial intelligence and machine learning technologies for integration of terrestrial in non-terrestrial networks. *IEEE Internet of Things Magazine*, 7(1), 28-33. doi: 10.1109/IOTM.001.2300190.
[18] Khan, M.A., Kumar, N., Mohsan, S.A., Khan, W.U., Nasralla, M.M., Alsharif, M.H., Żywiołek, J., & Ullah, I. (2022). Swarm of UAVs for network management in 6G: A technical review. *IEEE Transactions on Network and Service Management*, 20(1), 741-761. doi: 10.1109/TNSM.2022.3213370.
[19] Khan, S.K., Naseem, U., Siraj, H., Razzak, I., & Imran, M. (2021). The role of unmanned aerial vehicles and mmWave in 5G: Recent advances and challenges. *Transactions on Emerging Telecommunications Technologies*, 32(7), article number e4241. doi: 10.1002/ett.4241.
[20] Kourtis, M.A., Xilouris, G., Oikonomakis, A., Batistatos, M.C., & Chochliouros, I. (2023). Integration of drone connectivity in 5G: An examination of the OASEES framework. In *Proceedings of the IEEE future networks world forum* (pp. 1-6). Baltimore: IEEE. doi: 10.1109/FNWF58287.2023.10520379.
[21] Lins, S., Cardoso, K.V., Both, C.B., Mendes, L., De Rezende, J.F., Silveira, A., Linder, N., & Klautau, A. (2021). Artificial intelligence for enhanced mobility and 5G connectivity in UAV-based critical missions. *IEEE Access*, 9, 111792-111801. doi: 10.1109/ACCESS.2021.3103041.
[22] Lu, Y., Wen, W., Igorevich, K.K., Ren, P., Zhang, H., Duan, Y., Zhu, H., & Zhang, P. (2023). UAV ad hoc network routing algorithms in space-air-ground integrated networks: Challenges and directions. *Drones*, 7(7), article number 448. doi: 10.3390/drones7070448.
[23] Majumdar, P., Mitra, S., Bhattacharya, D., & Bhushan, B. (2024). Enhancing sustainable 5G powered agriculture 4.0: Summary of low power connectivity, internet of UAV things, AI solutions and research trends. *Multimedia Tools and Applications*. doi: 10.1007/s11042-024-19728-1.
[24] Mohsen, B.M. (2024). Ai-driven optimization of urban logistics in smart cities: Integrating autonomous vehicles and IoT for efficient delivery systems. *Sustainability*, 16(24), article number 11265. doi: 10.3390/su162411265.
[25] Posvistak, V., & Miroshnychenko, D. (2024). Usage of mobile network for remote drone. *Technologies and Engineering*, 25(4), 73-84. doi: 10.30857/2786-5371.2024.4.7.
[26] Qasim, N.H., & Jawad, A.M. (2024). 5G-enabled UAVs for energy-efficient opportunistic networking. *Heliyon*, 10(12), article number e32660. doi: 10.1016/j.heliyon.2024.e32660.
[27] Rovira-Sugranes, A., Razi, A., Afghah, F., & Chakareski, J. (2022). A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook. *Ad Hoc Networks*, 130, article number 102790. doi: 10.1016/j.adhoc.2022.102790.
[28] Sarkar, N.I., & Gul, S. (2023). Artificial intelligence-based autonomous UAV networks: A survey. *Drones*, 7(5), article number 322. doi: 10.3390/drones7050322.
[29] Shayea, I., Dushi, P., Banafaa, M., Rashid, R.A., Ali, S., Sarijari, M.A., Daradkeh, Y.I., & Mohamad, H. (2022). Handover management for drones in future mobile networks – a survey. Sensors, 22(17), article number 6424. doi: 10.3390/s22176424.
[30] Solanki, A., Tarar, S., Singh, S.P., & Tayal, A. (2022). *The internet of drones: AI applications for smart solutions*. London: CRC Press.
[31] Sufyan, A., Khan, K.B., Khashan, O.A., Mir, T., & Mir, U. (2023). From 5G to beyond 5G: A comprehensive survey of wireless network evolution, challenges, and promising technologies. *Electronics*, 12(10), article number 2200. doi: 10.3390/electronics12102200.
[32] Tlili, F., Ayed, S., & Fourati, L.C. (2024). Advancing UAV security with artificial intelligence: A comprehensive survey of techniques and future directions. *Internet of Things*, 27, article number 101281. doi: 10.1016/j.iot.2024.101281.
[33] Trabelsi, N., Fourati, L.C., & Chen, C.S. (2024). Interference management in 5G and beyond networks: A comprehensive survey. *Computer Networks*, 239, article number 110159. doi: 10.1016/j.comnet.2023.110159.
[34] Warrier, A., Al-Rubaye, S., Inalhan, G., & Tsourdos, A. (2023). AI-enabled interference mitigation for autonomous aerial vehicles in urban 5G networks. *Aerospace*, 10(10), article number 884. doi: 10.3390/aerospace10100884.
[35] Wu, Q., Xu, J., Zeng, Y., Ng, D.W., Al-Dhahir, N., Schober, R., & Swindlehurst, A.L. (2021). A comprehensive overview on 5G-and-beyond networks with UAVs: From communications to sensing and intelligence. *IEEE Journal on Selected Areas in Communications*, 39(10), 2912-2945. doi: 10.1109/JSAC.2021.3088681.

[36] Yermolenko, R., Klekots, D., & Gogota, O. (2024). Development of an algorithm for detecting commercial unmanned aerial vehicles using machine learning methods. *Machinery & Energetics,* 15(2), 33-45. doi: 10.31548/machinery/2.2024.33.
[37] Zaid, M., Kadir, M.K., Shayea, I., & Mansor, Z. (2024). Machine learning-based approaches for handover decision of cellular-connected drones in future networks: A comprehensive review. *Engineering Science and Technology, an International Journal*, 55, article number 101732. doi: 10.1016/j.jestch.2024.101732.

Інтеграція 5G та штучного інтелекту для вдосконалення маршрутизації у дронах

Роман Зайвий
Аспірант
Національний університет «Львівська політехніка»
79000, вул. Степана Бандери, 12, м. Львів, Україна
https://orcid.org/0009-0003-4096-4111
Олег Мельничок
Аспірант
Національний університет «Львівська політехніка»
79000, вул. Степана Бандери, 12, м. Львів, Україна
https://orcid.org/0009-0003-9766-9682

Анотація. Розвиток технологій 5G та штучного інтелекту створює нові можливості для вдосконалення маршрутизації безпілотних літальних апаратів, що є особливо актуальним для логістики, рятувальних операцій та моніторингу критичної інфраструктури. Метою дослідження був аналіз перспектив впровадження 5G та AI у маршрутизацію дронів, визначення ключових викликів та розробка рекомендацій для їх ефективної інтеграції у повітряний простір України. У дослідженні використано методи теоретичного аналізу наукових джерел, порівняльного аналізу міжнародного досвіду та систематизації сучасних підходів до маршрутизації дронів із використанням 5G і AI. Проведено аналіз архітектури 5G-мереж, алгоритмів оптимізації маршрутів та механізмів координації роїв дронів. Основні результати дослідження продемонстрували, що поєднання 5G та AI забезпечує значне підвищення ефективності автономних безпілотних систем, дозволяючи оперативно адаптувати маршрути, оптимізувати енергоспоживання та підвищувати рівень безпеки польотів. Особлива увага в дослідженні приділена порівнянню двох популярних алгоритмів оптимізації маршрутів для БПЛА: алгоритм рою часток та мурашиний алгоритм. Аналіз показав, що обидва алгоритми ефективно вирішують завдання маршрутизації, однак мають свої переваги в залежності від специфіки застосування. Алгоритм рою часток виявився більш ефективним для задач з великою кількістю змінних, дозволяючи оптимізувати маршрути в реальному часі при швидко змінюваних умовах. Мурашиний алгоритм, в свою чергу, продемонстрував перевагу у вирішенні складних задач з великою кількістю перешкод. Практичне значення дослідження полягає у визначенні ключових технічних і регуляторних викликів, пов'язаних з інтеграцією 5G та AI у маршрутизацію дронів, а також у розробці науково обґрунтованих підходів до їх вирішення. Отримані результати можуть бути використані для вдосконалення національних нормативних актів, сприяння впровадженню інтелектуальних безпілотних систем у сферах логістики, моніторингу інфраструктури та рятувальних операцій, а також для подальших досліджень у галузі автономних авіаційних технологій

Ключові слова: безпілотні літальні апарати; штучний інтелект; блокчейн; нейромережа; рої дронів

BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en Vol. 30 No. 1. 2025

UDC 004.89:519.816
DOI: 10.62660/bcstu/1.2025.91

Information technology for solving the multi-criteria decision-making problem using the modified Fuzzy TOPSIS method

Anton Maksymov
PhD Student
Cherkasy State Technological University
18006, 460 Shevchenka Blvd, Cherkasy, Ukraine
https://orcid.org/0009-0002-0411-8164

Yurii Tryus*
Doctor of Pedagogical Sciences, PhD in Physical and Mathematical Sciences, Professor
Cherkasy State Technological University
18006, 460 Shevchenka Blvd, Cherkasy, Ukraine
https://orcid.org/0000-0002-0739-2065

Abstract. The relevance of the research topic is determined by the need to effectively solve multi-criteria decision-making problems in conditions of fuzzy information. In this regard, the creation of information technologies that would enable the user to select and use the most effective multi-criteria decision-making methods in conditions of fuzzy information is an important problem. The purpose of the study was to develop information technology for solving the multi-criteria decision-making problem using the modified Fuzzy Technique for Order Preference by Similarity to Ideal Solution (FTOPSIS) method based on the use of different metrics and the results of group expertise, which increases the reliability of the obtained decisions. Within the framework of the study, an analysis of the most popular multi-criteria decision-making methods, in particular, methods using the fuzzy set apparatus, was carried out. The article analyses different popular metrics for estimating the distances between a fuzzy positive ideal solution and a fuzzy negative ideal solution in the FTOPSIS method. A technique is proposed for comparing the results of applying different methods, in particular, FTOPSIS using triangular and trapezoidal fuzzy numbers, TOPSIS with triangular and trapezoidal fuzzy numbers for determining criteria weights, which makes it possible to analyse the scale of deviations between the obtained results and to assess the quality of the experts' work. The obtained results expand the possibilities of using TOPSIS and FTOPSIS methods for decision-making in conditions of multi-criteriality and uncertainty. As a practical application of the developed information technology and the modified FTOPSIS method, the article solves the problem of selecting the best of popular risk management standards in IT projects. This will increase the effectiveness of risk management in conditions of uncertainty and incompleteness of information, improve the validity of decisions made, as well as adapt the risk management process to specific conditions of each individual IT project

Keywords: MCDM; MADM; fuzzy sets; FTOPSIS; IT projects; risk management; risk management standards

INTRODUCTION

The relevance of multi-criteria decision-making problems lies in the fact that in real conditions of management, planning and analysis, it is necessary to take into account a set of conflicting criteria that affect

Article's History: Received: 13.10.2024; Revised: 24.02.2025; Accepted: 17.03.2025.

Suggested Citation:
Maksymov, A., & Tryus, Yu. (2025). Information technology for solving the multi-criteria decision-making problem using the modified Fuzzy TOPSIS method. *Bulletin of Cherkasy State Technological University*, 30(1), 91-106. doi: 10.62660/bcstu/1.2025.91.

*Corresponding author

the choice of the best solution. This is especially important in conditions of uncertainty, when it is necessary to find a compromise between different performance indicators, such as costs, quality, risks and system stability. Multi-criteria analysis methods make it possible not only to formalise the selection process, but also ensure the validity of decisions, increasing their effectiveness and objectivity. The use of modern methods and information technologies in multi-criteria decision-making helps to automate the decision-making process, which is a key factor in complex and dynamic systems. The scientific community is actively researching and developing new modifications of the FTOPSIS method. In particular, Y. Kustiyahningsih *et al.* (2024) in their work analysed the use of different types of fuzzy logic in the FTOPSIS method, but exclusively using the Euclidean metric to determine the distance between an ideal and an anti-ideal solution.

N. Rane & S. Choudhary (2023) in their study conducted a comparative analysis of the effectiveness of the Fuzzy AHP and FTOPSIS methods, which also used the classical Euclidean metric. The results showed that both methods were effective tools for multi-criteria decision-making under uncertainty, but FTOPSIS demonstrated a slightly higher sensitivity to changes in weight coefficients of the criteria compared to Fuzzy AHP. H. Yin *et al.* (2020) in their work used the classical Euclidean metric to measure the distance between alternatives in the context of multi-criteria analysis using the TOPSIS method. The authors drew attention to the limitations of the classical Euclidean distance, which may not always correctly reflect the difference between fuzzy or unstructured data, which often arises in practical problems. In this regard, they proposed a modification of the Euclidean distance, which was called the relative Euclidean distance.

In their study, F. Han *et al.* (2024) demonstrated the practical application of the FTOPSIS method using the Euclidean distance for solving multi-criteria decision-making problems. This study demonstrated the practical use of FTOPSIS, particularly in the context of using standard metrics in real decision-making problems. In the study of P. Talukdar & P. Dutta (2019), six different distance metrics were considered within the framework of the classical TOPSIS method, which allowed for a comparative analysis based on examples. However, this study needs to be expanded in the direction of formalising a general model of using different metrics for the FTOPSIS method. In turn, H.-S. Shyur & H.-S. Shih (2024) analysed the impact of the choice of metric on the results of ranking alternatives in the classical TOPSIS method. The authors emphasise the importance of the correct choice of metric, since different metrics can significantly change the ranking results, affecting decision-making in multi-criteria problems.

The study of H. Arman *et al.* (2022) was focused on the incorrect use of the Euclidean metric to measure the difference between fuzzy numbers. The researchers emphasised that although the Euclidean metric was used to measure the distance between points in multidimensional spaces, its application to fuzzy numbers was conceptually incorrect. The authors concluded that the use of the Euclidean metric in the context of fuzzy numbers can lead to inaccurate results, since such a metric does not reflect their essential difference. Summarising the results of the analysis of scientific sources, it can be stated that the vast majority of existing studies are focused on the use of the standard Euclidean metric in the TOPSIS method and its fuzzy modification – FTOPSIS.

Thus, one of the tasks of this study is to analyse the impact of the choice of metric on the results of the FTOPSIS method. The involvement of experts in assessing alternatives according to different criteria is an important aspect of using multi-criteria decision-making methods based on the fuzzy set apparatus. This leads to the fact that the obtained fuzzy assessments, especially during individual expertise, are subjective in nature. In this situation, to improve the accuracy of the assessment of alternatives, it is advisable to use an expertise involving several experts. In turn, this leads to the need to apply methods of aggregating expert opinions. So, there is a need to develop information technologies that provide group expertise using fuzzy multi-criteria decision-making methods and implement the procedure for aggregating expert assessments.

Therefore, an urgent problem arises, which consists in developing a modified FTOPSIS method using appropriate adapted metrics, as well as creating information technology that implements them and takes into account the results of group expertise for a more accurate and effective assessment and choice of the best alternative from many possible ones according to several utility criteria. It is the solution to this problem that this article was devoted to.

MATERIALS AND METHODS

The multi-criteria decision-making problem is to assess and select the best alternative from the set of $A = \{A_1, A_2, \ldots, A_N\}$ possible options, taking into account the set of $C = \{C_1, C_2, \ldots, C_M\}$ criteria that characterise these alternatives. Some criteria can be benefit criteria (the higher the value, the better), let's denote the set of these criteria as $C = \{C_1, C_2, \ldots, C_M\}$, P is the number of benefit (maximisation) criteria ($0 \le P \le M$), and other criteria are cost criteria (the lower the value, the better), let's denote the set of these criteria as $C_{min} = \{C_{P+1}, C_{P+2}, \ldots, C_M\}$, while the number of cost (minimisation) criteria is equal to M-P. Each C_j criterion has its own w_j weight, while $w_j > 0, \sum_{j=1}^{M} w_j = 1$. The effectiveness of each A_i $(i = \overline{1,N})$ alternative is assessed relative to each C_j $(j = \overline{1,M})$ criterion using the value of the corresponding $f_j(A_i, w_j, x_{ij})$ utility function, where x_{ij} – the assessment of the value of the C_j criterion relative to the A_i alternative taking into account the $w_j (i = \overline{1,N}, j = \overline{1,M})$ weight. Formally, the MCDM problem can be written as follows:

$$\begin{cases} \max\limits_{A_i \in A} f_j\left(A_i, w_j, x_{ij}\right), j = \overline{1,P}, \\ \min\limits_{A_i \in A} f_j\left(A_i w_j, x_{ij}\right), j = \overline{P+1,M}. \end{cases} \quad (1)$$

Then the $A^* \in A$ alternative, which satisfies the condition:

$$A^* = \begin{cases} arg\max\limits_{A_i \in A} f_j\left(A_i, w_j, x_{ij}\right), j = \overline{1,P}, \\ arg\min\limits_{A_i \in A} f_j\left(A_i, w_j, x_{ij}\right), j = \overline{P+1,M}. \end{cases} \quad (2)$$

is the solution to the problem (1).

The A^* alternative of the form (2) can be considered as an ideal solution to the problem (1), which in practice can only be found approximately.

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), proposed by C. Hwang & K. Yoon (1981), is one of the most well-known methods for an approximate solution to the multi-criteria decision-making problem of the form (1). This method is based on the concept that the chosen alternative should have the smallest distance to the positive ideal solution (PIS) (a solution that maximises benefit criteria and minimises cost criteria) and the largest distance to the negative ideal solution (NIS) (a solution that minimises benefit criteria and maximises cost criteria). Linguistic assessments are a convenient and effective tool for reflecting expert assessments, especially in cases where precise quantitative information is unavailable or uncertain. They allow experts to express their judgments in an understandable form, which is often based on natural language.

In their works, C. Chen (2000) and C. Chen *et al.* (2006) extended the TOPSIS method by a procedure that uses linguistic assessments in the form of triangular and trapezoidal fuzzy numbers (FNs). Let's consider some aspects of this approach. In general, the parametric form of a triangular fuzzy number is a triple (*a*, *b*, *c*), where *a* - left, *b* - middle, *c* - right parametres of this fuzzy number, and the parametric form of a trapezoidal fuzzy number is a quadruple (*a*, *b*, *c*, *d*), where *a* - left, *b* - left middle, *c* - right middle, *d* - right parametres of this fuzzy number. Examples of linguistic assessments and their representation in the form of triangular and trapezoidal fuzzy numbers are given in Table 1.

Table 1. Examples of linguistic assessments for fuzzy numbers

Linguistic terms	Fuzzy triangular numbers	Fuzzy trapezoidal numbers
Very low (VL)	(1, 1, 3)	(1, 1, 2, 3)
Low (L)	(1, 3, 5)	(2, 3, 4, 5)
Middle (M)	(3, 5, 7)	(4, 5, 6, 7)
High (H)	(5, 7, 9)	(6, 7, 8, 9)
Very high (VH)	(7, 9, 9)	(8, 9, 10, 10)

Source: created by the authors based on C. Chen (2000), C. Chen *et al.* (2006)

Linguistic assessments are linguistic terms to which certain fuzzy numbers correspond, usually in a triangular or trapezoidal parametric form (Fig. 1).

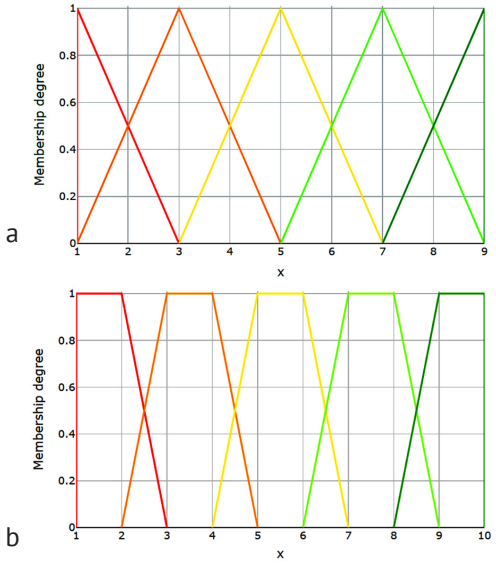

Figure 1. Examples of fuzzy triangular numbers (a) and trapezoidal numbers (b)

Source: created by the authors based on C. Chen *et al.* (2006)

To calculate the distance from each alternative to the fuzzy positive ideal solution (FPIS) and the fuzzy negative ideal solution (FNIS) in FTOPSIS, C. Chen (2000) proposed a vertex method for calculating the distance between two fuzzy numbers, the essence of which is as follows. Let $\tilde{x}=(a_1,a_2,\ldots,a_n)$, $\tilde{y}=(b_1,b_2,\ldots,b_n)$ be fuzzy numbers specified by their parametres, where $n=3$ for triangular FNs and $n=4$ for trapezoidal FNs. Then the generalised adapted formula for calculating the Euclidean distance between triangular and trapezoidal FNs has the form:

$$d_E(\tilde{x},\tilde{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)^2}. \quad (3)$$

Let's consider the features of calculating the distances between triangular and trapezoidal FNs using some popular metrics.

The Manhattan metric is a metric according to which the distance between two points $x=(x_1, x_2,\ldots, x_n)$ and $y=(y_1, y_2,\ldots, y_n)$ is equal to the sum of the moduli of the difference of their components (Rudin, 1987; Deza & Deza, 2016; Black, 2019):

$$d_{MH}(x,y) = \sum_{i=1}^{n} |x_i - y_i|. \quad (4)$$

Let $\tilde{x} = (a_1, a_2,\ldots, a_n)$, $\tilde{y} = (b_1, b_2,\ldots, b_n)$ be fuzzy numbers specified by their parametres, where $n = 4$ for

triangular FNs and for trapezoidal FNs. Then the generalised adapted formula (4) for triangular and trapezoidal FNs has the form:

$$d_{MH}(\tilde{x},\tilde{y}) = \sum_{i=1}^{n}|a_i - b_i|, \quad (5)$$

where $n=3$ for triangular FNs; $n=4$ for trapezoidal FNs.

The Chebyshev metric is a metric that determines the distance between two -dimensional numerical vectors $x=(x_1, x_2,\dots, x_n)$ and $y=(y_1, y_2,\dots, y_n)$ as the maximum of the moduli of the difference of their components (Deza & Deza, 2016). The Chebyshev metric is often denoted by l_∞:

$$l_\infty(\vec{x},\vec{y}) = \max_{i=\overline{1,n}}|x_i - y_i|. \quad (6)$$

Then the generalised adapted formula (6) for FNs $\tilde{x}=(a_1, a_2,\dots, a_n)$, $\tilde{y}=(b_1, b_2,\dots, b_n)$ has the form:

$$d_\infty(\tilde{x},\tilde{y}) = \max_{1\le i\le n}|a_i - b_i|, \quad (7)$$

where $n=3$ for triangular FNs; $n=4$ for trapezoidal FNs.

The Minkowski metric is a parametric metric that can be considered as a generalisation of the Euclidean metric and the Manhattan metric. The Minkowski distance of p order between two points $x=(x_1, x_2,\dots, x_n)$ and $y=(y_1, y_2,\dots, y_n)$ is determined by the formula:

$$l_p(x,y) = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}. \quad (8)$$

The Minkowski metric is a generalisation of the Euclidean metric for the $p=2$ parametre, and when $p=1$, the Manhattan metric is obtained (Deza & Deza, 2016). If p goes to plus infinity, then the Minkowski distance approaches the Chebyshev distance:

$$\lim_{p\to+\infty}\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} = \max_{i=\overline{1,n}}|x_i - y_i|. \quad (9)$$

Similarly, when p goes to minus infinity, the following formula is obtained:

$$\lim_{p\to-\infty}\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} = \min_{i=\overline{1,n}}|x_i - y_i|. \quad (10)$$

The generalised adapted formula (8) for FNs $\tilde{x}=(a_1, a_2,\dots, a_n)$, $\tilde{y}=(b_1, b_2,\dots, b_n)$ has the form:

$$d_M(\tilde{x},\tilde{y}) = \left(\frac{1}{n}\sum_{i=1}^{n}|a_i - b_i|^p\right)^{\frac{1}{p}}, \quad (11)$$

where $n=3$ for triangular FNs; $n=4$ for trapezoidal FNs.

The Hamming metric is a metric that calculates the number of positions in which the corresponding symbols of two words of the same length are different. The Hamming distance between two points $x=(x_1, x_2,\dots, x_n)$ and $y=(y_1, y_2,\dots, y_n)$ is determined by the formula:

$$d_H(\tilde{x},\tilde{y}) = \sum_{i=1}^{n}|x_i - y_i|, \text{ where } |x_i - y_i| = \begin{cases}1, & x_i \ne y_j\\ 0, & x_i = y_j\end{cases}. \quad (12)$$

The generalised adapted formula (12) for FNs $\tilde{x}=(a_1, a_2,\dots, a_n)$, $\tilde{y}=(b_1, b_2,\dots, b_n)$ has the form:

$$d_H(\tilde{x},\tilde{y}) = \sum_{i=1}^{n}|a_i - b_i|, \text{ where } |a_i - b_i| = \begin{cases}1, & a_i \ne b_i\\ 0, & a_i = b_i\end{cases}, \quad (13)$$

where $n=3$ for triangular FNs; $n=4$ for trapezoidal FNs.

Let's consider the main steps of the method, which is a modification of the FTOPSIS method for the case of group expertise and uses different metrics for calculating the distance from each alternative to FPIS and FNIS in order to determine the best alternative.

Suppose that there is a group of decision-makers consisting of K experts.

Step 1. Assignment of fuzzy assessments to criteria and alternatives.

Each k-th expert of the $(k=\overline{1,K})$ group determines the linguistic assessment of the weight of each C_j $(j=\overline{1,M})$ criterion to which a triangular fuzzy number $\tilde{w}_j^k = \left(w_{j1}^k, w_{j2}^k, w_{j3}^k\right)$ corresponds. Next, the k-th expert of the group determines the linguistic assessment for each A_i $(i=\overline{1,N})$ alternative with respect to the C_j $(j=\overline{1,M})$ criterion to which the triangular fuzzy number $\tilde{x}_{ij}^k = \left(a_{ij}^k, b_{ij}^k, c_{ij}^k\right)$ corresponds.

If fuzzy trapezoidal numbers $\tilde{w}_j^k$ and $\tilde{x}_{ij}^k$ correspond to linguistic assessments of the k-th expert of the group, then they have the following form, respectively:

$$\tilde{w}_j^k = \left(w_{j1}^k, w_{j2}^k, w_{j3}^k, w_{j4}^k\right),\ \tilde{x}_{ij}^k = \left(a_{ij}^k, b_{ij}^k, c_{ij}^k, d_{ij}^k\right). \quad (14)$$

Step 2. Calculation of aggregated fuzzy weights for criteria and aggregated fuzzy assessments for alternatives.

The aggregated fuzzy weight assessment in triangular form $\tilde{w}_j = \left(w_{j1}, w_{j2}, w_{j3}\right)$ for C_j criterion is calculated according to the formulas:

$$w_{j1} = \min_{k=\overline{1,K}}\left\{w_{j1}^k\right\},$$
$$w_{j2} = \frac{1}{K}\sum_{k=1}^{K} w_{j2}^k\ ,\ w_{j3} = \max_{k=\overline{1,K}}\left\{w_{j3}^k\right\}, j=\overline{1,M}. \quad (15)$$

Similarly, the aggregated triangular fuzzy assessment $\tilde{x}_{ij}=(a_{ij}, b_{ij}, c_{ij})$ for A_i alternative with respect to C_j criterion is calculated according to the formulas:

$$a_{ij} = \min_{k=\overline{1,K}}\left\{a_{ij}^k\right\},$$
$$b_{ij} = \frac{1}{K}\sum_{k=1}^{K} b_{ij}^k\ ,\ c_{ij} = \max_{k=\overline{1,K}}\left\{c_{ij}^k\right\}, i=\overline{1,N},\ j=\overline{1,M}. \quad (16)$$

These formulas make it possible to obtain a common fuzzy assessment that takes into account minimum, average and maximum values from fuzzy assessments of all experts in the group.

Let $\tilde{w}_j^k = \left(w_{j1}^k, w_{j2}^k, w_{j3}^k, w_{j4}^k\right)$ be the trapezoidal fuzzy weight assessment determined by the k-th expert in the group for C_j $(j=\overline{1,M})$ criterion. Then the aggregated fuzzy weight $\tilde{w}_j = \left(w_{j1}, w_{j2}, w_{j3}, w_{j4}\right)$ for C_j criterion can be calculated according to the following formulas:

$$w_{j1} = \min_{k=\overline{1,K}}\left\{w_{j1}^k\right\},\ w_{j2} = \frac{1}{K}\sum_{k=1}^{K} w_{j2}^k\ ,$$
$$w_{j3} = \frac{1}{K}\sum_{k=1}^{K} w_{j3}^k\ ,\ w_{j4} = \max_{k=\overline{1,K}}\left\{w_{j4}^k\right\},\ j=\overline{1,M} \quad (17)$$

Let $\tilde{x}_{ij}^{k} = \left(a_{ij}^{k}, b_{ij}^{k}, c_{ij}^{k}, d_{ij}^{k}\right)$ be the trapezoidal fuzzy assessment of the k-th expert of the decision-making group for A_i $(i = \overline{1,N})$ alternative with respect to C_j $(j = \overline{1,M})$ criterion. Then the aggregated fuzzy assessments for trapezoidal fuzzy numbers $\tilde{x}_{ij} = \left(a_{ij}, b_{ij}, c_{ij}, d_{ij}\right)$ are determined according to the formulas: $x\ \tilde{r}_{ij}$

$$a_{ij} = \min_{k=\overline{1,K}}\{a_{ij}^{k}\}, b_{ij} = \frac{1}{K}\sum_{k=1}^{K} b_{ij}^{k}, c_{ij} = \frac{1}{K}\sum_{k=1}^{K} c_{ij}^{k},$$
$$d_{ij} = \max_{k=\overline{1,K}}\{d_{ij}^{k}\}, i = \overline{1,N}, j = \overline{1,M}. \quad (18)$$

Step 3. Calculation of the normalised fuzzy decision matrix.

The normalised fuzzy decision matrix is denoted as $\tilde{R} = [\tilde{r}_{ij}]$, in which for triangular FNs:

$$\tilde{r}_{ij} = \left(r_{ij1}, r_{ij2}, r_{ij3}\right) = \left(\frac{a_{ij}}{c_j^{+}}, \frac{b_{ij}}{c_j^{+}}, \frac{c_{ij}}{c_j^{+}}\right), \quad i = \overline{1,N}, j = \overline{1,P}, \quad (19)$$

where $c_j^{+} = \max_{i=\overline{1,N}}\{c_{ij}\}$ – for the benefit criterion, and

$$\tilde{r}_{ij} = \left(r_{ij1}, r_{ij2}, r_{ij3}\right) = \left(\frac{a_j^{-}}{c_{ij}}, \frac{a_j^{-}}{b_{ij}}, \frac{a_j^{-}}{a_{ij}}\right), \quad i = \overline{1,N}, j = \overline{P+1,M}, \quad (20)$$

where $a_j^{-} = \min_{i=\overline{1,N}}\{a_{ij}\}$ - for the cost criterion.

In the normalised fuzzy matrix for trapezoidal FNs it will be:

$$\tilde{r}_{ij} = \left(r_{ij1}, r_{ij2}, r_{ij3}, r_{ij4}\right) = \left(\frac{a_{ij}}{d_j^{+}}, \frac{b_{ij}}{d_j^{+}}, \frac{c_{ij}}{d_j^{+}}, \frac{d_{ij}}{d_j^{+}}\right), \quad i = \overline{1,N}, j = \overline{1,P}, \quad (21)$$

where $d_j^{+} = \max_{i=\overline{1,N}}\{d_{ij}\}$ – for the benefit criterion, and

$$\tilde{r}_{ij} = \left(r_{ij1}, r_{ij2}, r_{ij3}, r_{ij4}\right) = \left(\frac{a_j^{-}}{d_{ij}}, \frac{a_j^{-}}{c_{ij}}, \frac{a_j^{-}}{b_{ij}}, \frac{a_j^{-}}{a_{ij}}\right), \quad i = \overline{1,N}, j = \overline{P+1,M}, \quad (22)$$

where $a_j^{-} = \min_{i=\overline{1,N}}\{a_{ij}\}$ – for the cost criterion.

Step 4. Calculation of the weighted normalised fuzzy decision matrix.

Let's calculate the weighted normalised fuzzy decision matrix

$$\tilde{V} = \left(\tilde{v}_{ij}\right), i = \overline{1,N}, j = \overline{1,M}, \quad (23)$$

where by for triangular fuzzy assessments

$$\tilde{v}_{ij} = \left(v_{ij1}, v_{ij2}, v_{ij3}\right) = \left(r_{ij1} \cdot w_{j1}, r_{ij2} \cdot w_{j2}, r_{ij3} \cdot w_{j3}\right), \quad (24)$$

and for trapezoidal fuzzy assessments

$$\tilde{v}_{ij} = \left(v_{ij1}, v_{ij2}, v_{ij3}, v_{ij4}\right) = \left(r_{ij1} \cdot w_{j1}, r_{ij2} \cdot w_{j2}, r_{ij3} \cdot w_{j3}, r_{ij4} \cdot w_{j4}\right). \quad (25)$$

Step 5. Calculation of the fuzzy positive ideal solution (FPIS) and the fuzzy negative ideal solution (FNIS).

The values of FPIS and FNIS are calculated according to the formulas:

$$V^{+} = (\tilde{v}_1^{+}, \tilde{v}_2^{+}, \cdots, \tilde{v}_M^{+}), \text{where } \tilde{v}_j^{+} = \max_{i=\overline{1,N}}\{v_{ijm}\}\ j = \overline{1,M}, \quad (26)$$

$$V^{-} = (\tilde{v}_1^{-}, \tilde{v}_2^{-}, \cdots, \tilde{v}_M^{-}), \text{where } \tilde{v}_j^{-} = \min_{i=\overline{1,N}}\{v_{ij1}\}\ j = \overline{1,M}, \quad (27)$$

where $m = 3$ for triangular FNs; $m = {}^{4}3$ for trapezoidal FNs.

Step 6. Calculation of distances from each alternative to FPIS and FNIS.

The distance from A_i alternative to FPIS d_i^{+} and to FNIS d_i^{-} is calculated according to the formulas:

$$d_i^{+} = \sum_{j=1}^{M} d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right), \quad (28)$$

$$d_i^{-} = \sum_{j=1}^{M} d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right), i = \overline{1,N}, \quad (29)$$

where $d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right)$ and $d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right)$ according to the user's choice are determined as follows:

- for the Euclidean metric, taking into account (3):

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = = d_E\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(v_{ijk} - v_{jk}^{+}\right)^2}, \quad (30)$$

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = = d_E\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(v_{ijk} - v_{jk}^{-}\right)^2}, \quad (31)$$

- for the Manhattan metric, taking into account (5):

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = = d_{MH}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = \sum_{k=1}^{n}\left|v_{ijk} - v_{jk}^{+}\right|, \quad (32)$$

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = = d_{MH}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = \sum_{k=1}^{n}\left|v_{ijk} - v_{jk}^{-}\right|, \quad (33)$$

- for the Chebyshev metric, taking into account (7):

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = = d_{\infty}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = \max_{k=\overline{1,n}}\left|v_{ijk} - v_{jk}^{+}\right|, \quad (34)$$

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = = d_{\infty}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = \max_{k=\overline{1,n}}\left|v_{ijk} - v_{jk}^{-}\right|, \quad (35)$$

- for the Minkowski metric, taking into account (11):

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = d_M\left(\tilde{v}_{ij}, \tilde{v}_j^{+}\right) = \left(\frac{1}{n}\sum_{k=1}^{n}\left|v_{ijk} - v_{jk}^{+}\right|^{p}\right)^{\frac{1}{p}}, \quad (36)$$

$$d_{metrics}\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = = d_M\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right) = \left(\frac{1}{n}\sum_{k=1}^{n}\left|v_{ijk} - v_{jk}^{-}\right|^{p}\right)^{\frac{1}{p}}, \quad (37)$$

at $p \geq 3$

• for the Hamming metric, taking into account (13):

$$d_{metrics}(\tilde{v}_{ij}, \tilde{v}_j^+) = \\ = d_H(\tilde{v}_{ij}, \tilde{v}_j^+) = \sum_{k=1}^{n} |v_{ijk} - v_{jk}^+|, \quad (38)$$

$$d_{metrics}(\tilde{v}_{ij}, \tilde{v}_j^-) = \\ = d_H(\tilde{v}_{ij}, \tilde{v}_j^-) = \sum_{k=1}^{n} |v_{ijk} - v_{jk}^-|, \quad (39)$$

where

$$|v_{ijk} - v_{jk}^+| = \begin{cases} 1, & v_{ijk} \neq v_{jk}^+ \\ 0, & v_{ijk} = v_{jk}^+, \end{cases} \quad (40)$$

$$|v_{ijk} - v_{jk}^-| = \begin{cases} 1, & v_{ijk} \neq v_{jk}^- \\ 0, & v_{ijk} = v_{jk}^-, \end{cases} \quad (41)$$

$n = 3$ for triangular FNs and $n = 4$ for trapezoidal FNs.

Step 7. Calculation of the closeness coefficients to FPIS and FNIS for each alternative according to chosen metrics.

The CC_i^- closeness coefficient for each A_i alternative to the fuzzy negative ideal solution is calculated using the following formula:

$$CC_i^- = \frac{d_i^-}{d_i^- + d_i^+}, i = \overline{1, N}. \quad (42)$$

The CC_i^+ closeness coefficient for each A_i alternative to the fuzzy positive ideal solution is calculated using the following formula:

$$CC_i^+ = \frac{d_i^+}{d_i^- + d_i^+}, i = \overline{1, N}. \quad (43)$$

Step 8. Ranking of alternatives according to chosen metrics. In this step, for each chosen metric, the best A^* alternative is determined as the alternative that is furthest from FNIS, i.e. with the maximum CC_i^- closeness coefficient, and is defined as follows:

$$A^* = A_{i^*}, \quad (44)$$

where

$$i^* = \arg \max_{i=\overline{1,N}} \{CC_i^-\}, \quad (45)$$

or A^* is selected as the alternative that is closest to FPIS, i.e. with the smallest CC_i^+ closeness coefficient, and is defined as:

$$A^* = A_{i^*}, \quad (46)$$

where

$$i^* = \arg \min_{i=\overline{1,N}} \{CC_i^+\}. \quad (47)$$

Therefore, the larger the CC_i^- value, the closer the A_i alternative is to the fuzzy positive ideal solution V^+ and further from the fuzzy negative ideal solution V^-, or the smaller the CC_i^+ value, the closer the A_i alternative is to the fuzzy positive ideal solution V^+ and further from the fuzzy negative ideal solution V^-. The A_i alternatives are ordered according to decrease in CC_i^- values, or according to increase in CC_i^+ values, which makes it possible to determine the rank of each R_i $(i = \overline{1, N})$ alternative for chosen metrics. In this case, some alternatives may have the same ranks, including rank 1.

Step 9. Output of results. After calculating the ranks of alternatives according to chosen metrics, the results are presented in tabular form, the structure of which is given in Table 2, where R_{ji} – the rank of A_i $(i = \overline{1, N})$ alternative according to the $d_{metricsj}$ $j = \overline{1, m}$ metric, m – the number of chosen metrics.

Table 2. Output of the results of the modified FTOPSIS method

Metrics	Alternatives	d_i^+	d_i^-	CC_i^-	Rank of alternatives
$d_{metrics_1}$	A_1	d_{11}^+	d_{11}^-	CC_{11}^-	R_{11}
	A_2	d_{12}^+	d_{12}^-	CC_{12}^-	R_{12}
	...	...	...	...	...
	A_N	d_{1N}^+	d_{1N}^-	CC_{1N}^-	R_{1N}
$d_{metrics_2}$	A_1	d_{21}^+	d_{21}^-	CC_{21}^-	R_{21}
	A_2	d_{22}^+	d_{22}^-	CC_{22}^-	R_{22}
	...	...	...	...	...
	A_N	d_{2N}^+	d_{2N}^-	CC_{2N}^-	R_{2N}
...	...	...	...	...	...
$d_{metrics_m}$	A_1	d_{m1}^+	d_{m1}^-	CC_{m1}^-	R_{m1}
	A_2	d_{m2}^+	d_{m2}^-	CC_{m2}^-	R_{m2}
	...	...	...	...	...
	A_N	dm_N^+	dm_N^-	CC_{mN}^-	R_{mN}

Source: created by the authors

For the software implementation of the proposed modified Fuzzy TOPSIS method, the authors created a module that is an integral part of the web-oriented Decision Support System (DSS) "Decisioner" (n.d.). In the developed DSS, the PHP language is used to create queries to the MySQL database. This programming language is also used to distribute access between users depending on their role. The information and

registration part of the DSS was created using the CMS WordPress. The LaTeX data markup language was used to display formulas in the theoretical information of the methods. The JavaScript language without frameworks was used to implement the decision-making methods in the DSS. Actually, in order for JavaScript and PHP to interact with the user, standard HTML5 and CSS3 tools were chosen for visual design. The JQuery library was used for the interaction of algorithms of the methods with the HTML code of the page. The amCharts and Plotly libraries were used to display graphs.

RESULTS AND DISCUSSION

Multi-Criteria Decision Making (MCDM) is divided into two main categories:

- Multi-Objective Decision Making (MODM) is an approach to optimisation, in which a solution is chosen from a set of possible alternatives by finding a compromise between several objective functions. This approach is widely used in linear and non-linear programming problems, in particular for optimisation of production processes, resource allocation and management of complex systems under conditions of multi-criteriality;
- Multi-Attribute Decision Making (MADM) is an approach focused on choosing the best alternative from a discrete set of solutions based on several attributes (criteria). This is useful for problems where it is necessary to assess and rank a finite number of options.

Therefore, MODM is usually used in optimisation problems, and MADM is used to assess and choose effective alternatives. Both approaches are important components of MCDM and are used depending on the specific problem and available input data.

The analysis of modern research, such as F. Han *et al.* (2024), shows that among MADM methods, considerable attention is paid to methods based on fuzzy set theory, in particular the FTOPSIS method and the FAHP method. At the same time, the FTOPSIS method is more effective compared to the FAHP method in a number of situations due to several key aspects. FTOPSIS is convenient for the practical application, since it focuses on finding the most ideal solution that is as close as possible to the ideal (best) option and as far away as possible from the worst option. This makes the method easy to understand and apply, especially when it is necessary to easily and quickly assess alternatives according to several criteria. FAHP, in turn, often requires more computational steps and deeper analysis, including a hierarchy of criteria and sub-criteria, which can be more difficult in the case of a large number of criteria.

FTOPSIS works well with large numbers of alternatives and criteria. This allows for rapid assessment within large data sets. FTOPSIS does not require the construction of a complex hierarchy of criteria, which simplifies the decision-making process, especially in cases where complex relationships between criteria are not obvious. FAHP can become difficult with large numbers of criteria and/or alternatives, as it requires the construction of a hierarchical structure and the comparison of each pair of criteria according to the main criterion, as well as the comparison of each pair of alternatives according to each criterion, which quickly increases the amount of calculations. FAHP requires careful construction of the hierarchy structure of the research object and important comparisons between each level of the hierarchy, which can be difficult and time-consuming. In general, FTOPSIS is effective in cases, where it is necessary to quickly and efficiently assess alternatives according to a large number of criteria.

Most modern approaches to assessing distances in TOPSIS and FTOPSIS methods (Awasthi *et al.*, 2011; Arman *et al.*, 2022) are usually based on the Euclidean metric. However, in the context of complex and multidimensional data, this metric does not always adequately reflect the real differences between alternatives. This emphasises the need to study and implement alternative metrics, such as the Manhattan metric, Chebyshev, Minkowski and Hamming metrics. However, the use of alternative metrics in the context of fuzzy sets requires additional adaptation. Most classical metrics were developed to work with precise numerical data and do not take into account the peculiarities of representing fuzzy data, for example, in the form of triangular or trapezoidal fuzzy numbers. This complicates the direct use of standard formulas for calculating distances between alternative assessments. Therefore, it is necessary to develop new metrics or modify existing metrics so that they can correctly process fuzzy data, ensuring accuracy and reliability of the assessment. The study proposes a modified FTOPSIS method, which provides an opportunity to find effective alternatives according to the most popular metrics adapted to calculations with fuzzy numbers.

Practical application of the developed information technology and the modified Fuzzy TOPSIS method is demonstrated on the problem of choosing a risk management standard in IT projects based on their analysis, given in the author's article of A. Maksymov (2025). Let the decision makers need to determine the most effective risk management standard to be used by the IT company from the following alternatives: A_1 – use of ISO 31000:2018 (2018); A_2 – use of NIST 800-53 (National Institute of Standards and Technology Special Publication 800-53, 2020); A_3 – use of the PMBOK guide (2021). In this case, it is proposed to be guided by the following C_j criteria for assessing the effectiveness of choosing a risk management standard ($j = \overline{1,8}$):

- C_1 – number of countries implementing the standard (measured in quantity and to be maximised);
- C_2 – impact on the number of incidents/violations (measured in percentage and to be maximised);
- C_3 – impact on productivity (measured in percentage and to be maximised);
- C_4 – impact on employee satisfaction level (measured in percentage and to be maximised);

- C_5 – impact on financial performance (measured in percentage and to be maximised);
- C_6 – implementation costs (measured in thousands of $ and to be minimised);
- C_7 – implementation time (measured in months and to be minimised);
- C_8 – number of training hours for personnel (measured in hours and to be minimised).

For example, the results of an individual expertise of the problem are given. Based on the authors' analysis of risk management standards, linguistic assessments of the weights of the C_j $(j = \overline{1,8})$ criteria were determined, as well as for each A_i $(i = \overline{1,3})$ alternative according to each C_j criterion, both explicit values and intervals of utility, and their linguistic assessments given in Table 3 were determined.

Table 3. Assessments of different risk management standards for IT projects according to the specified criteria

	C_1 (countries)	C_2 (%)	C_3 (%)	C_4 (%)	C_5 (%)	C_6 (k$)	C_7 (months)	C_8 (hours)
Weight	Medium (M)	High (H)	High (H)	High (H)	Very High (VH)	Very High (VH)	High (H)	Medium (M)
min/max	max	max	max	max	max	min	min	min
A_1	100+ Very High (VH)	10-30 Medium (M)	10-20 High (H)	75-85 High (H)	10-15 Medium (M)	5-20 Medium (M)	3-6 Medium (M)	20-40 Medium (M)
A_2	50+ High (H)	20-40 High (H)	15-25 Very High (VH)	70-80 High (H)	20-30 Very High (VH)	50-150 Very High (VH)	6-12 High (H)	40-80 High (H)
A_3	100+ Very High (VH)	30-50 High (H)	20-30 Very High (VH)	80-90 High (H)	15-20 High (H)	20-100 High (H)	6-12 High (H)	40-60 High (H)

Source: created by the authors

To determine the best alternative among the risk management standards for IT projects according to certain criteria, calculations were first carried out using the modified FTOPSIS method by the appropriate *DSS "Decisioner"* module. Figure 2 demonstrates the first step in solving the problem, in particular, filling in the appropriate fields of the module with information about the purpose of the problem and its structural components: alternatives and criteria.

In the next step, it is necessary to select metrics and the type of fuzzy numbers for calculation, for example, as shown in Figure 3.

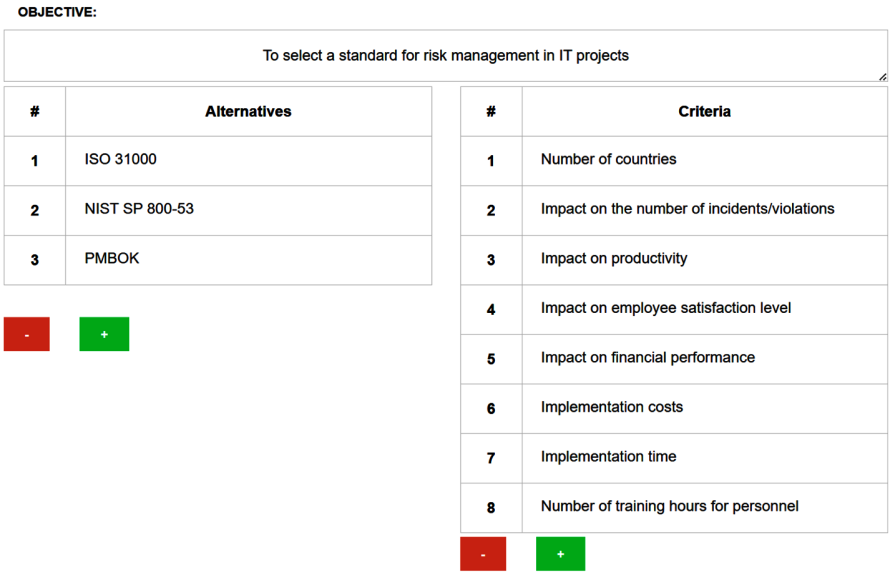

Figure 2. Setting the goal, alternatives and criteria in the DSS "Decisioner"

Source: created by the authors

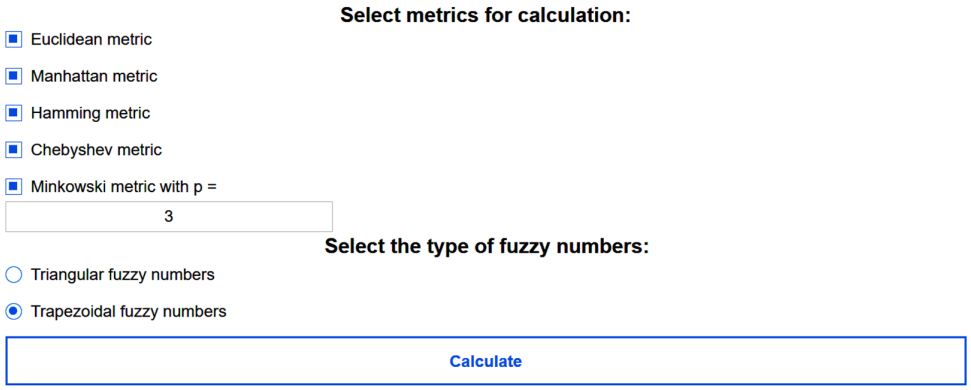

Figure 3. Selection of metrics and type of fuzzy numbers for calculation
Source: created by the authors

In the next step, the expert needs to fill in the initial matrix of assessments in the format of fuzzy linguistic assessments, for example, based on the data in Table 1 (Fig. 4).

	Number of countries	Impact on the number of incidents/violations	Impact on productivity	Impact on employee satisfaction level	Impact on financial performance	Implementation costs	Implementation time	Number of training hours for personnel
min/max	max	max	max	max	max	min	min	min
Weight	M	H	H	H	VH	VH	H	M
ISO 31000	VH	M	H	H	M	M	M	M
NIST SP 800-53	H	H	VH	H	VH	VH	H	H
PMBOK	VH	H	VH	H	H	H	H	H

Figure 4. Filling in the initial matrix of assessments in the format of linguistic terms
Source: created by the authors

After confirming the input data, the DSS automatically calculates the normalised fuzzy decision matrix $\tilde{R} = [\tilde{r}_{ij}]$ of the form (19)-(20) for triangular FNs, or of the form (21)-(22) for trapezoidal FNs (Fig. 5).

	C1	C2	C3	C4	C5	C6	C7	C8
min/max	max	max	max	max	max	min	min	min
ISO 31000	0.8000 0.9000 1.0000 1.0000	0.4444 0.5556 0.6667 0.7778	0.6000 0.7000 0.8000 0.9000	0.6667 0.7778 0.8889 1.0000	0.4000 0.5000 0.6000 0.7000	0.5714 0.6667 0.8000 1.0000	0.5714 0.6667 0.8000 1.0000	0.5714 0.6667 0.8000 1.0000
NIST SP 800-53	0.6000 0.7000 0.8000 0.9000	0.6667 0.7778 0.8889 1.0000	0.8000 0.9000 1.0000 1.0000	0.6667 0.7778 0.8889 1.0000	0.8000 0.9000 1.0000 1.0000	0.4000 0.4000 0.4444 0.5000	0.4444 0.5000 0.5714 0.6667	0.4444 0.5000 0.5714 0.6667
PMBOK	0.8000 0.9000 1.0000 1.0000	0.6667 0.7778 0.8889 1.0000	0.8000 0.9000 1.0000 1.0000	0.6667 0.7778 0.8889 1.0000	0.6000 0.7000 0.8000 0.9000	0.4444 0.5000 0.5714 0.6667	0.4444 0.5000 0.5714 0.6667	0.4444 0.5000 0.5714 0.6667

Figure 5. Normalised fuzzy decision matrix for trapezoidal FNs
Source: created by the authors

In the next step, the DSS automatically calculates the weighted normalised fuzzy decision matrix $\tilde{V} = [\tilde{v}_{ij}]$ of the form (23)-(25) (Fig. 6).

Figure 7 shows the result of calculating the best A^* alternative with a display of the distances from each A_i alternative to FPIS (d_i^+) of the form (28) and FNIS (d_i^-) of the form (29) for selected metrics, as well as the CC_i^- closeness coefficients of the form (42) ($i = \overline{1,3}$) and R_{ji} – the rank of A_i ($i = \overline{1,N}$) alternative according to the $d_{metrics_j}$ ($j = \overline{1,m}$) metric, m – the number of selected metrics.

	C1 (max)	C2 (max)	C3 (max)	C4 (max)	C5 (max)	C6 (min)	C7 (min)	C8 (min)
A1	3.2000 4.5000 6.0000 7.0000	2.6667 3.8889 5.3333 7.0000	3.6000 4.9000 6.4000 8.1000	4.0000 5.4444 7.1111 9.0000	3.2000 4.5000 6.0000 7.0000	4.5714 6.0000 8.0000 10.0000	3.4286 4.6667 6.4000 9.0000	2.2857 3.3333 4.8000 7.0000
A2	2.4000 3.5000 4.8000 6.3000	4.0000 5.4444 7.1111 9.0000	4.8000 6.3000 8.0000 9.0000	4.0000 5.4444 7.1111 9.0000	6.4000 8.1000 10.0000 10.0000	3.2000 3.6000 4.4444 5.0000	2.6667 3.5000 4.5714 6.0000	1.7778 2.5000 3.4286 4.6667
A3	3.2000 4.5000 6.0000 7.0000	4.0000 5.4444 7.1111 9.0000	4.8000 6.3000 8.0000 9.0000	4.0000 5.4444 7.1111 9.0000	4.8000 6.3000 8.0000 9.0000	3.5556 4.5000 5.7143 6.6667	2.6667 3.5000 4.5714 6.0000	1.7778 2.5000 3.4286 4.6667
V+	3.2000 4.5000 6.0000 7.0000	4.0000 5.4444 7.1111 9.0000	4.8000 6.3000 8.0000 9.0000	4.0000 5.4444 7.1111 9.0000	6.4000 8.1000 10.0000 10.0000	4.5714 6.0000 8.0000 10.0000	3.4286 4.6667 6.4000 9.0000	2.2857 3.3333 4.8000 7.0000
V-	2.4000 3.5000 4.8000 6.3000	2.6667 3.8889 5.3333 7.0000	3.6000 4.9000 6.4000 8.1000	4.0000 5.4444 7.1111 9.0000	3.2000 4.5000 6.0000 7.0000	3.2000 3.6000 4.4444 5.0000	2.6667 3.5000 4.5714 6.0000	1.7778 2.5000 3.4286 4.6667

Figure 6. Weighted normalised fuzzy decision matrix for trapezoidal FNs

Source: created by the authors

Metrics	Alternatives	d_i^+	d_i^-	CC_i^-	Rank of alternatives
Euclidean metric	A1 A2 A3	7.4563 8.8192 8.2978	8.8192 7.4563 8.0161	0.5419 0.4581 0.4914	1 3 2
Manhattan metric	A1 A2 A3	8.5222 9.2767 8.7794	9.2767 8.5222 9.0196	0.5212 0.4788 0.5067	1 3 2
Hamming metric	A1 A2 A3	12.0000 16.0000 16.0000	16.0000 12.0000 20.0000	0.5714 0.4286 0.5556	1 3 2
Chebyshev metric	A1 A2 A3	2.5333 3.8444 3.5556	3.8444 2.5333 2.8222	0.6028 0.3972 0.4425	1 3 2
Minkowski metric with p = 3	A1 A2 A3	7.1764 9.0394 8.4769	9.0394 7.1764 7.7997	0.5574 0.4426 0.4792	1 3 2

Answer: The best alternative corresponds to the option with a ranking of "1"

Figure 7. Result of calculating the best alternative for trapezoidal FNs

Source: created by the authors

Based on the results obtained according to the FTOPSIS method for different metrics, ISO 31000:2018 (2018) (A_1 alternative) according to all selected metrics is the best option for the risk management standard for IT projects among the proposed alternatives. The results of the experiment on solving the problem with the help of different modifications of the FTOPSIS method and the classical TOPSIS method with linguistic assessments of criteria weights are shown below:

1) FT3FNs – FTOPSIS using triangular FNs (see Fig. 1 a);

2) FT4FNs – FTOPSIS using trapezoidal FNs (see Fig. 1 b);

3) TFW3FNs – TOPSIS with fuzzy triangular numbers for determining criteria weights;

4) TFW4FNs – TOPSIS with fuzzy trapezoidal numbers for determining criteria weights.

In the TFW3FNs, TFW4FNs methods, in order to simplify the expert assessment of w_j, $(j = \overline{1,M})$ criteria weights, which in sum must be equal to one, in the classical TOPSIS method, linguistic assessments of the weights of each C_j criterion to which fuzzy numbers $\widetilde{w}_j = (w_{j1}, w_{j2}, \dots, w_{jn})$, $(j = \overline{1,M})$, correspond, where $n = 3$ for triangular FNs and $n = 4$ for trapezoidal FNs, were used. The process of defuzzification of fuzzy numbers was carried out according to the following rule:

1. Finding of criteria weights as the average value of fuzzy assessment parametres.

For each $\widetilde{w}_j = (w_{j1}, w_{j2}, \dots, w_{jn})$ the arithmetic mean of its parametres is calculated:

$$w_j = \frac{w_{j1}+w_{j2}+\cdots+w_{jn}}{n}, j = \overline{1,M}, \quad (48)$$

where $n = 3$ for triangular FNs and $n = 4$ for trapezoidal FNs.

2. Calculation of the sum of average values:

$$ws = \sum_{j=1}^{M} w_j. \quad (49)$$

3. Normalisation of average values of the weights. Each w_j average value is converted into a fraction of the total sum "*ws*" so that all normalised assessments as a result are in the [0,1] range and their sum is equal to 1:

$$wn_j = \frac{w_j}{ws}, j = \overline{1,M}. \quad (50)$$

Fig. 8 shows a table with problem input data for the TOPSIS method with linguistic assessments of criteria weights to which trapezoidal FNs correspond (see Table 1).

Fig. 9 shows a table with problem input data for the TOPSIS method with normalised assessments of criteria weights according to formulas (48)-(50) for $n = 4$.

	Number of countries	Impact on the number of incidents/violations	Impact on productivity	Impact on employee satisfaction level	Impact on financial performance	Implementation costs	Implementation time	Number of training hours for personnel
min/max	max	max	max	max	max	min	min	min
Weight	M	H	H	H	VH	VH	H	M

Figure 8. Problem input data for the TOPSIS method with linguistic assessments of criteria weights
Source: created by the authors

	Number of countries	Impact on the number of incidents/violations	Impact on productivity	Impact on employee satisfaction level	Impact on financial performance	Implementation costs	Implementation time	Number of training hours for personnel
min/max	max	max	max	max	max	min	min	min
Weight	0.09243(	0.12605042016806	0.12605042	0.1260504:	0.15546218‹	0.15546218487	0.1260504201(	0.092436
ISO 31000	100	20	15	80	12.5	12.5	4.5	30
NIST SP 800-53	50	30	20	75	25	100	9	60
PMBOK	100	40	25	85	17.5	60	9	50

Figure 9. Problem input data for the TOPSIS method with normalised assessments of criteria weights
Source: created by the authors

In order to visually display the differences between the results when using different metrics and methods, a matrix of deviations between the results obtained by different methods was calculated according to such technique. Let there be a set of $M_1, M_2, \ldots, M_n$ methods, each of which gives closeness coefficients of each A_i $(i = \overline{1,N})$ alternative according to different metrics in the form of a matrix:

$$E^q = \begin{pmatrix} CC_{11}^q & CC_{12}^q & \ldots & CC_{1m}^q \\ CC_{21}^q & CC_{22}^q & \ldots & CC_{2m}^q \\ \vdots & \vdots & \ddots & \vdots \\ CC_{N1}^q & CC_{N2}^q & \ldots & CC_{Nm}^q \end{pmatrix}, \quad (51)$$

where $q = \overline{1,n}$ – the number of the M_q method, CC_{ij}^q – the values of closeness coefficients for A_i $(i = \overline{1,N})$ alternatives, obtained according to the M_q method using the metric with the j $(j = \overline{1,m})$ number, while both CC_i^- coefficients of the form (42) and CC_i^+ ones of the form (43) can be chosen as closeness coefficients. Then the $D^{X,Y}$ matrix of deviations between the results of two methods X and Y $(X = \overline{1,n-1}, Y = \overline{2,n}, X \neq Y)$ is calculated according to the formula:

$$D^{X,Y} = \begin{pmatrix} |CC_{11}^X - CC_{11}^Y| & |CC_{12}^X - CC_{12}^Y| & \ldots & |CC_{1m}^X - CC_{1m}^Y| \\ |CC_{21}^X - CC_{21}^Y| & |CC_{22}^X - CC_{22}^Y| & \ldots & |CC_{2m}^X - CC_{2m}^Y| \\ \vdots & \vdots & \ddots & \vdots \\ |CC_{N1}^X - CC_{N1}^Y| & |CC_{N2}^X - CC_{N2}^Y| & \ldots & |CC_{Nm}^X - CC_{Nm}^Y| \end{pmatrix}, \quad (52)$$

where CC_{ij}^x and CC_{ij}^y – the values of closeness coefficients in the corresponding methods X and Y, N – the number of alternatives, n – the number of methods, m – the number of metrics in each of applied methods.

Let M_1 be FT3FNs, M_2 – FT4FNs, M_3 – TFW3FNs and M_4 – TFW4FNs. After conducting an experiment with the considered methods with different j $(j = \overline{1,5})$ metrics for A_i $(i = \overline{1,3})$ alternatives, E^q $(q = 1,4)$ matrices of the form (51) shown in Fig. 10 were obtained.

$D^{X,Y}$ deviation matrices of the form (52) between the results of two methods X and Y $(X = \overline{1,3}, Y = \overline{2,4}, X \neq Y)$ are presented in Fig. 11.

	E^1	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
FT3FNs	A1	0,5764	0,5387	0,5789	0,6261	0,5970
	A2	0,4236	0,4613	0,4211	0,3739	0,4030
	A3	0,4524	0,4787	0,5217	0,4062	0,4354
	E^2	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
FT4FNs	A1	0,5419	0,5212	0,5714	0,6028	0,5574
	A2	0,4581	0,4788	0,4286	0,3972	0,4426
	A3	0,4914	0,5067	0,5556	0,4425	0,4792
	E^3	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
TFW3FNs	A1	0,6087	0,6055	0,5556	0,6631	0,6250
	A2	0,3282	0,2690	0,3000	0,3369	0,3363
	A3	0,5103	0,5656	0,6364	0,4571	0,4936
	E^4	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
TFW4FNs	A1	0,6119	0,6082	0,5556	0,6631	0,6282
	A2	0,3280	0,2689	0,3000	0,3369	0,3361
	A3	0,5083	0,5639	0,6364	0,4571	0,4914

Figure 10. Results of methods in the form of E^q matrix of the form (51)
Source: created by the authors

	$D^{1,2}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|FT3FNs - FT4FNs|	A1	0,0345	0,0175	0,0075	0,0234	0,0396
	A2	0,0345	0,0175	0,0075	0,0234	0,0396
	A3	0,0389	0,0280	0,0338	0,0363	0,0438
	$D^{1,3}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|FT3FNs - TFW3FNs|	A1	0,0324	0,0667	0,0234	0,0370	0,0280
	A2	0,0955	0,1922	0,1211	0,0370	0,0667
	A3	0,0579	0,0868	0,1146	0,0509	0,0582
	$D^{1,4}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|FT3FNs - TFW4FNs|	A1	0,0355	0,0694	0,0234	0,0370	0,0312
	A2	0,0957	0,1923	0,1211	0,0370	0,0669
	A3	0,0559	0,0851	0,1146	0,0509	0,0559
	$D^{2,3}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|FT4FNs - TFW3FNs|	A1	0,0669	0,0843	0,0159	0,0603	0,0676
	A2	0,1300	0,2098	0,1286	0,0603	0,1063
	A3	0,0190	0,0588	0,0808	0,0146	0,0144
	$D^{2,4}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|FT4FNs - TFW4FNs|	A1	0,0700	0,0870	0,0159	0,0603	0,0707
	A2	0,1302	0,2099	0,1286	0,0603	0,1064
	A3	0,0170	0,0571	0,0808	0,0146	0,0122
	$D^{3,4}$	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
|TFW3FNs - TFW4FNs|	A1	0,003134289	0,002700502	0	0	0,00311769
	A2	0,000179553	9,22555E-05	0	0	0,000158067
	A3	0,002005483	0,001686576	0	0	0,002290157

Figure 11. $D^{X,Y}$ deviation matrices of the form (52) between the results of the methods
Source: created by the authors

As can be seen from Figure 11, the deviations are unevenly distributed, which means that different metrics respond differently to the same input data. The smallest deviation between closeness coefficients is obtained in the $D^{3,4}$ deviation matrix (Fig. 11), which indicates that there is no significant difference in the choice of the type of fuzzy numbers (triangular or trapezoidal FNs) for determining criteria weights. The largest deviation in $D^{3,4}$ is 0.0027 in the Manhattan metric. Also, insignificant deviations between closeness coefficients are observed in the $D^{1,2}$ deviation matrix (Fig. 11). Therefore, it can be concluded that there is no significant difference between the use of FTOPSIS using triangular and trapezoidal FNs. The largest deviation in $D^{1,2}$ is 0.0438 in the Minkowski metric. Relatively significant deviations between closeness coefficients are obtained in $D^{1,3}$, $D^{1,4}$, $D^{2,3}$ and $D^{2,4}$ deviation matrices. In particular, the largest deviations are observed when using the Manhattan metric (on average 0.2) for the above deviation matrices. This is due to the peculiarity of the calculation of this metric.

As a result of the experiments, the Chebyshev metric is determined to be the most stable metric (Fig. 11). The proposed approach for comparing the results of applying different methods makes it possible to determine which methods give similar or sharply different results. Using this approach, it is possible to analyse whether the scale of deviations is significant or whether it is within the limits of permissible error. Also, this approach to comparing the results of the use of different methods provides an opportunity to assess the quality of the experts' work, namely, how accurately the experts have carried out the fuzzification of real input data to the decision-making problem. If it is necessary to select a universal metric, then it is advisable to use metrics with minimal deviations (for example, the Minkowski or Chebyshev metric). If it is important to take into account individual maximum differences between FPIS and FNIS, then the Manhattan metric may be better. Methods that have minimal deviations between the results can be used alternatively, and methods that have significant deviations between the results require additional analysis in order to identify and explain the reasons for the obtained discrepancies. To compare the results of choosing the best alternative, a matrix of comparison of the ranks of alternatives is considered (Fig. 12).

FT3FNs	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
A1	1	1	1	1	1
A2	3	3	3	3	3
A3	2	2	2	2	2
FT4FNs	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
A1	1	1	1	1	1
A2	3	3	3	3	3
A3	2	2	2	2	2
TFW3FNs	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
A1	1	1	2	1	1
A2	3	3	3	3	3
A3	2	2	1	2	2
TFW4FNs	Euclidean	Manhattan	Hamming	Chebyshev	Minkowski
A1	1	1	2	1	1
A2	3	3	3	3	3
A3	2	2	1	2	2

Figure 12. Matrix of comparison of the ranks of alternatives

Source: created by the authors

According to the results of applying the FT3FNs, FT-4FNs, TFW3FNs, TFW4FNs methods with different metrics (Fig. 12), it was found that ISO 31000:2018 (2018) (A_1 alternative) is the best option for the risk management standard for IT projects among the proposed alternatives. Within the framework of this study, a modified FTOPSIS method, which involves the use of adapted metrics that more correctly reflect the differences between fuzzy numbers, is proposed. In addition, the implemented information technology takes into account the results of group expertise, which provides increased accuracy in the process of multi-criteria selection. The conducted study is consistent with a number of previous works, which also emphasise the importance of metric selection in the context of fuzzy data. Thus, N. Rane & S. Choudhary (2023) and A. Makki & R. Abdulaal (2023) demonstrated the effectiveness of using FTOPSIS, but used the standard Euclidean distance.

Also, Y. Kustiyahningsih *et al.* (2024) demonstrated the use of FTOPSIS with the classical Euclidean metric, which, despite its prevalence, has a number of limitations in case of fuzzy or non-standardised data. H. Arman *et al.* (2022) critically analysed the feasibility of using the Euclidean metric in FTOPSIS, proving its conceptual inadequacy for comparing fuzzy numbers. In turn, H. Yin *et al.* (2020) proposed an alternative – relative Euclidean distance – that better takes into account the features of fuzzy data. The results

obtained in P.Talukdar & P.Dutta (2019) and H.-S.Shyur & H.S. Shih (2024) confirm that the choice of metric can significantly affect the result of ranking alternatives. However, these studies do not focus on the fuzzy environment, which limits their relevance in the context of modern decision-making problems with a high level of uncertainty.

Based on the analysis of advantages and disadvantages of the analysed MADM methods, a modification of the FTOPSIS method is proposed using various metrics, in particular, the Euclidean metric, the Manhattan metric, the Chebyshev, Minkowski and Hamming metrics to estimate the closeness coefficient to fuzzy positive and fuzzy negative ideal solutions. Thus, the comparative analysis shows that the modification of the FTOPSIS method proposed in this work, taking into account the adapted metrics and group expertise, makes it possible to eliminate a number of shortcomings of previous approaches, providing a more accurate, flexible and substantiated assessment of alternatives in a multi-criteria environment. In the conducted scientific research:

- a modified FTOPSIS method is proposed using the results of group expertise and adaptation of popular metrics for triangular and trapezoidal fuzzy numbers, which ensures the preservation of metrics properties and allows to more accurately assess the distances between fuzzy ideal and anti-ideal solutions;
- an approach is proposed for determining criteria weights in the classical TOPSIS method through the use of linguistic assessments, which allows for more flexible consideration of uncertainty and subjectivity of expert assessments and is important for real decision-making scenarios;
- an approach is proposed for comparing the results of applying different methods, in particular FTOPSIS with the use of triangular and trapezoidal fuzzy numbers, TOPSIS with triangular and trapezoidal fuzzy numbers obtained using different metrics, which provides an opportunity to analyse the scale of deviations and assess the quality of experts' work.

CONCLUSIONS

The study analysed the most popular multi-criteria decision-making methods, in particular, such methods as TOPSIS, FTOPSIS and FAHP. Particular attention was paid to the analysis of different popular metrics for determining the distances between fuzzy positive and fuzzy negative ideal solutions in the FTOPSIS method. Based on the results of this analysis, a modified Fuzzy TOPSIS method for selecting an effective alternative in the multi-criteria decision-making problem based on different metrics and using the results of group expertise was proposed. On the example of a real problem of choosing a risk management standard in IT projects, it was demonstrated how the use of different metrics affects the accuracy and reliability of choosing an effective solution. An approach is proposed for comparing the results of applying different methods, in particular, the FTOPSIS methods with the use of triangular and trapezoidal fuzzy numbers, TOPSIS with triangular and trapezoidal fuzzy numbers for determining criteria weights obtained by different metrics for estimating the closeness coefficients to a fuzzy positive ideal solution (FPIS) and a fuzzy negative ideal solution (FNIS).

The results of the study open up new opportunities for applying the TOPSIS and FTOPSIS methods in various areas of decision-making, especially in conditions of multi-criteriality and uncertainty. The developed web-oriented application that implements the proposed modified Fuzzy TOPSIS method can also be used to solve multi-criteria decision-making problems in various fields of human activity under conditions of fuzzy data. Prospects for further research include improving methods for aggregating expert opinions, expanding the range of metrics used to increase the assessment accuracy, as well as adapting the proposed approach to dynamic decision-making conditions, in particular in real time and taking into account changes in criteria priorities. It is also relevant to study the possibilities of integrating the proposed approach with other approaches based on fuzzy logic, such as neural-fuzzy systems and hybrid models based on machine learning.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Arman, H., Hadi-Vencheh, A., Kiani Mavi, R., Khodadadipour, M., & Jamshidi, A. (2022). Revisiting the interval and fuzzy TOPSIS methods: Is Euclidean distance a suitable tool to measure the differences between fuzzy numbers? *Complexity,* 2022, article number 7032662. doi: 10.1155/2022/7032662.

[2] Awasthi, A., Chauhan, S.S., & Goyal, S.K. (2011). A multi-criteria decision making approach for location planning for urban distribution centers under uncertainty. *Mathematical and Computer Modelling,* 53(1-2), 98-109. doi: 10.1016/j.mcm.2010.07.023.

[3] Black, P.E. (2019). *Manhattan distance*. Retrieved from https://xlinux.nist.gov/dads/HTML/manhattanDistance.html.

[4] Chen, C.T. (2000). Extension of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems,* 114(1), 1-9. doi: 10.1016/S0165-0114(97)00377-1.

[5] Chen, C.T., Lin, C.T., & Huang, S.F. (2006). A fuzzy approach for supplier evaluation and selection in supply chain management. *International Journal of Production Economics*, 102(2), 289-301. doi: 10.1016/j.ijpe.2005.03.009.
[6] Decisioner. (n.d.). Retrieved from https://dss.knsa.chdtu.edu.ua/.
[7] Deza, M.M., & Deza, E. (2016). *Encyclopedia of distances*. Berlin: Springer. doi: 10.1007/978-3-662-52844-0.
[8] Han, F., Alkhawaji, R., & Shafieezadeh, M. (2024). Evaluating sustainable water management strategies using TOPSIS and fuzzy TOPSIS methods. *Applied Water Science,* 15, article number 4. doi: 10.1007/s13201-024-02336-7.
[9] Hwang, C.L., & Yoon, K.P. (1981). *Multiple attribute decision making. Methods and applications. A state-of-the-art survey*. Berlin: Springer. doi: 10.1007/978-3-642-48318-9.
[10] ISO 31000:2018. (2018). *Risk management – guidelines*. Retrieved from https://www.iso.org/iso-31000-risk-management.html.
[11] Kustiyahningsih, Y., Rahmanita, E., Khusnul Khotimah, B., & Purnama, J. (2024). Type-2 fuzzy ANP and TOPSIS methods based on trapezoid fuzzy number with a new metric. *International Journal of Advances in Intelligent Informatics,* 10(2), article number 239. doi: 10.26555/ijain.v10i2.1285.
[12] Makki, A., & Abdulaal, R. (2023). A hybrid MCDM approach based on fuzzy MEREC-G and fuzzy RATMI. *Mathematics*, 11(17), article number 3773. doi: 10.3390/math11173773.
[13] Maksymov, A. (2025). Analysis of risk management standards and their application in IT projects. *Management of Development of Complex Systems*, 61, 66-75. doi: 10.32347/2412-9933.2025.61.66-75,
[14] National Institute of Standards and Technology Special Publication 800-53. (2020). Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf.
[15] PMBOK guide. (2021). Retrieved from https://www.pmi.org/standards/pmbok.
[16] Rane, N., & Choudhary, S. (2023). Fuzzy AHP and fuzzy TOPSIS as an effective and powerful multi-criteria decision-making (MCDM) method for subjective judgments in selection process. *International Research Journal of Modernization in Engineering Technology and Science,* 5, 3786-3799. doi: 10.56726/IRJMETS36629.
[17] Rudin, W. (1987). *Real and complex analysis*. Singapore: McGraw-Hill.
[18] Shyur, H.-J., & Shih, H.-S. (2024). Resolving rank reversal in TOPSIS: A comprehensive analysis of distance metrics and normalization methods. *Informatica*, 35(4), 837-858. doi: 10.15388/24-INFOR576.
[19] Talukdar, P., & Dutta, P. (2019). A comparative study of TOPSIS method via different distance measure. *International Journal of Research in Advent Technology,* 7(5), 118-126. doi: 10.32622/ijrat.75201937.
[20] Yin, H., Li, X., & Gao, Y. (2020). Relative Euclidean distance with application to TOPSIS and estimation performance ranking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2), 1052-1064. doi: 10.1109/TSMC.2020.3017814.

Інформаційна технологія розв'язання задачі багатокритеріального прийняття рішень модифікованим методом Fuzzy TOPSIS

Антон Максимов

Аспірант
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0009-0002-0411-8164

Юрій Триус

Доктор педагогічних наук, кандидат фізико-математичних наук, професор
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
https://orcid.org/0000-0002-0739-2065

Анотація. Актуальність теми дослідження зумовлена потребою ефективного розв'язання задач багатокритеріального прийняття рішень в умовах нечіткої інформації. У зв'язку з цим важливою проблемою є створення інформаційних технологій, які б надавали можливість користувачу обирати і застосовувати найбільш ефективні методи багатокритеріального прийняття рішень в умовах нечіткої інформації. Метою дослідження була розробка інформаційної технології розв'язання задачі багатокритеріального прийняття рішень модифікованим методом Fuzzy Technique for Order Preference by Similarity to Ideal Solution (FTOPSIS) на основі використання різних метрик і результатів групової експертизи, що підвищує достовірність отриманих рішень. У межах дослідження проведено аналіз найпопулярніших методів багатокритеріального прийняття рішень, зокрема методів, що використовують апарат нечітких множин. У статті проведено аналіз різних популярних метрик для оцінки відстаней між нечітким позитивним ідеальним розв'язком та нечітким негативним ідеальним розв'язком в методі FTOPSIS. Запропоновано методику для порівняння результатів застосування різних методів, зокрема FTOPSIS із застосуванням трикутних та трапецієвидних нечітких чисел, TOPSIS з трикутними та трапецієвидними нечіткими числами для визначення ваг критеріїв, що надає можливість проаналізувати масштаби відхилень між одержаними результатами та оцінити якість роботи експертів. Отримані результати розширюють можливості використання методів TOPSIS і FTOPSIS для прийняття рішень в умовах багатокритеріальності та невизначеності. Як практичне застосування розробленої інформаційної технології і модифікованого методу FTOPSIS у статті розв'язується задача вибору найкращого з популярних стандартів управління ризиками в IT-проєктах. Це дозволить підвищити ефективність ризик-менеджменту в умовах невизначеності та неповноти інформації, підвищити обґрунтованість прийнятих рішень, а також адаптувати процес управління ризиками до специфічних умов кожного окремого IT-проєкту

Ключові слова: MCDM; MADM; нечіткі множини; FTOPSIS; IT-проєкти; управління ризиками; стандарти управління ризиками

ВІСНИК
Черкаського державного технологічного університету

Науковий збірник

Том 30, № 1. 2025

Заснований у 1996 р. Виходить чотири рази на рік

Оригінал-макет видання виготовлено у редакційно-видавничому відділі
Черкаського державного технологічного університету

Відповідальний редактор:
А. Лавданський

Підписано до друку 17 березня 2025 р.
Формат 60*84/8
Умов. друк. арк. 12,6
Наклад 50 прим.

Адреса видавництва:
Черкаський державний технологічний університет
18006, бульв. Шевченка, 460, м. Черкаси, Україна
Тел.: +380638013283
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/uk

BULLETIN
of Cherkasy State Technological University

Scientific Collection

Volume 30, No. 1. 2025

Founded in 1996. Published four times per year

The original layout of the publication is made in the editorial and publishing department
of Cherkasy State Technological University

Managing editor:
A. Lavdanskyi

Signed for print of March 17, 2025.
Format 60*84/8
Conventional printed pages 12.6
Circulation 50 copies

Editors Office Address:

Cherkasy State Technological University
18006, 460 Shevchenko Blvd., Cherkasy, Ukraine
Tel.: +380638013283
E-mail: info@bulletin-chstu.com.ua
https://bulletin-chstu.com.ua/en