**REPUBLIC OF AZERBAIJAN**

*On the rights of the manuscript*

# ABSTRACT

of the dissertation for the degree of Doctor of Philosophy

# DEVELOPMENT OF INTELLIGENT ANALYSIS METHODS AND ALGORITHMS FOR MALWARE DETECTION

| | |
|---|---|
| Speciality: | 3339.01 – "Information protection methods and systems, information security" |
| Field of science: | Technical Sciences |
| Applicant: | **Elshan Oruc oglu Baghirov** |

**Bakı – 2026**

The work was performed at the Institute of Information Technology of the Ministry of Science and Education of the Republic of Azerbaijan.

Scientific supervisor: Doctor of Technical Sciences, Assoc. Prof. **Yadigar Nasib oglu Imamverdiyev**

Official opponents: Doctor of Technical Sciences, Professor **Javanshir Firudin oglu Mammadov**

Doctor of Technical Sciences, Professor **Ramin Rza oglu Rzayev**

Doctor of Philosophy in Technical Sciences **Lala Hekayat gizi Karimova**

Dissertation Council FD 1.35 of Supreme Attestation Commission under the President of the Republic of Azerbaijan operating at the Institute of Information Technology of the Ministry of Science and Education.

Chairman of the Dissertation Council:
Full member of ANAS,
Doctor of Technical Sciences, Prof.
**Rasim Mahammad oglu Alguliyev**

Scientific secretary of the Dissertation council:
Doctor of Technical Sciences, Assoc.Prof.
**Zarifa Gasim gizi Jabrayilova**

Chairman of the scientific seminar:
Doctor of Technical Sciences
**Mutallim Mirzaahmed oglu Mutallimov**

## GENERAL CHARACTERISTICS OF THE WORK

**Relevance of the topic.** The advancement of information technologies and the widespread adoption of digital systems have increased the scale and complexity of information security threats. New types of malware and increasingly sophisticated evasion techniques place new demands on traditional protection mechanisms. Therefore, developing intelligent analysis methods and algorithms for timely and reliable malware detection is of significant scientific and practical importance.

Important steps have recently been taken in Azerbaijan within the framework of cyber defence. One such step is the "Strategy of the Republic of Azerbaijan on Information Security and Cybersecurity for 2023–2027," approved by the Decree of the President of the Republic of Azerbaijan dated 28 August 2023[1]. One of the targets set out in the strategy is to improve the technical, cryptographic, software and other means of information protection. One objective established to achieve this target is to ensure sustained organizational support at the state level for scientific research and experimental development in information security and cybersecurity.

Malware encompasses all types of software created and used to perform unauthorized and primarily harmful activities in computer systems and digital environments[2].

Other factors necessitating research into malware detection are listed below:

*1. The exponential increase in the number of malware samples, including newly emerging types of malware, as evidenced by statistical data***:** According to AV-TEST, more than 450,000 malware samples are registered every day[3].

*2. The severe destructive consequences of malware***:** In addition to

[1] "Azərbaycan Respublikasının informasiya təhlükəsizliyi və kibertəhlükəsizliyə dair 2023–2027-ci illər üçün Strategiyası"nın təsdiq edilməsi haqqında Azərbaycan Respublikası Prezidentinin Sərəncamı // 28 avqust 2023-cü il, № 4060. – Bakı, – 2023.

[2] Əliquliyev, R.M. İnformasiya təhlükəsizliyi insidentləri / R.M. Əliquliyev, Y.N. İmamverdiyev. – Bakı: İnformasiya Texnologiyaları, – 2012. – 219 s.

[3] AV-TEST. AV-ATLAS: Malware statistics: [Electronic resource]. – URL: https://portal.av-atlas.org/malware?lang=en.

disrupting certain user operations and slowing down computer performance, malware may result in substantial financial losses and even human casualties in critical infrastructure environments.

*3. Existing problems in the detection of malware by antivirus systems:* Currently, many traditional commercial antivirus systems heavily rely on static signatures—unique pattern identifiers used to recognize malicious software—and can consequently detect only the specific malware samples already indexed in their databases. Before a newly emerging or obfuscated malware sample can be analyzed and detected, at least one vulnerable computer system may already have been severely compromised.

*4. The role of software engineering in digital system security:* The increasing complexity of digital systems places greater demands on software security. Although security principles are incorporated throughout the lifecycle, from design to operation, the risk of malware infiltrating systems cannot be completely eliminated. Therefore, alongside resilient architectures, developing intelligent analysis methods and algorithms for timely malware detection remains a priority.

*5. The potential for applying intelligent analysis methods to malware detection:* Rapid technological advancements have significantly accelerated research dedicated to automated malware detection. By integrating advanced artificial intelligence paradigms—specifically machine learning and deep learning—across diverse domains, researchers have successfully designed novel detection frameworks that substantially outperform conventional rule-based and signature-based approaches.

**Object and subject of the research.** The object of the research is malware detection, while its subject is the development of intelligent analysis methods capable of determining whether software is malicious.

**The aim and issues of the work**. The aim of the research is to develop intelligent analysis methods and algorithms for the highly accurate detection of malware. For this purpose, the following issues were set in the dissertation work:

- To systematically analyse existing scientific research and

detection approaches in the field of malware detection, comparatively evaluate their operating principles, advantages, and limitations, and identify existing scientific and technological gaps;

- To develop a conceptual model reflecting the main stages and components of the malware detection process, as well as the interrelationships among them;
- To develop a method for selecting informative features obtained through static and dynamic analysis based on a reverse engineering approach, with the aim of minimising the computational time required for malware detection;
- To develop malware detection methods based on machine learning algorithms for solving malware classification and clustering problems and to evaluate their effectiveness;
- To develop a method based on deep learning architectures for the highly accurate detection of malware;
- To develop a method aimed at enriching training data through the use of Generative Adversarial Networks, reducing class imbalance, and improving the accuracy and generalisation capability of the malware detection model;
- To analyse the decision-making mechanisms of the proposed malware detection methods in terms of explainability and to experimentally evaluate their robustness, reliability, and adaptability to changing real-world conditions.

**Research methods.** To address the research objectives, machine learning techniques, including classification and clustering methods, autoencoders, Generative Adversarial Networks, and deep learning methods, were employed.

**The main provisions of the defense**:

- A hybrid filtering method for selecting informative features from the feature set generated through the static and dynamic analysis of malware, taking into account the discrete and continuous nature of the features;
- Binary and multiclass classification methods for detecting malware and determining the class to which it belongs, as well as a clustering method that enables malware samples to be

grouped on the basis of similar characteristics;

- A detection method based on representing malware binary files as images and processing these images using deep learning algorithms;
- A method for generating synthetic samples using Conditional Generative Adversarial Networks, reducing class imbalance in the dataset, and improving the accuracy of the malware detection model;
- An experimental approach for evaluating the explainability and robustness of the proposed malware detection methods and assessing their adaptability to changing real-world conditions.

**Scientific novelty of the research.** The following scientific results were obtained within the framework of the dissertation:

- A hybrid filtering method was developed for selecting informative features from the feature set generated through static and dynamic analysis, taking into account the discrete and continuous nature of the features;
- Binary and multiclass classification methods were developed for detecting malware and determining the class to which it belongs;
- A clustering method was developed for grouping malware samples whose class membership is difficult to determine in advance on the basis of similar features and for detecting previously unseen malware groups;
- A highly accurate detection method was developed based on converting malware binary files into image representations and analysing the resulting images using deep learning architectures;
- A method was developed for generating synthetic malware samples belonging to different classes using Conditional Generative Adversarial Networks, reducing class imbalance in the dataset, and improving the generalisation capability of the detection model;
- A model has been developed and experimentally validated to assess the integration of the proposed methods into operational

environments, their performance under changing data conditions, and their robustness.

**Theoretical and practical significance of the research.** *The theoretical significance of the dissertation* lies in enriching the scientific and methodological foundations of malware detection through an analysis of international practices and the development of a three-stage conceptual model and a hybrid filtering method that support accurate and robust machine learning models. *The practical significance of the work* lies in the direct applicability of the proposed algorithms to malware detection in Windows and Android environments and on mobile devices, as well as to real-time threat blocking and monitoring in security operations centers.

**Approbation of the research results.** The principal theoretical and practical results of the dissertation were presented at a number of international and national conferences listed below:

- The 1st (Baku, 2020), 2nd (Baku, 2021), and 4th (Baku, 2023) International Scientific Conferences of Students and Young Researchers;
- The 15th (Baku, 2021) and 17th (Baku, 2023) International Conferences on Application of Information and Communication Technologies (AICT);
- The 25th National Scientific Conference (NASCO) of PhD Students and Young Researchers, Baku, 2023;
- The 17th International Scientific and Technical Conference "Optoelectronic Devices in Image Recognition and Image Processing Systems," Kursk, 2023;
- The International Conference on Information Security: Problems and Prospects, Baku, 2022;
- The National Supercomputer Forum (NSCF), Pereslavl-Zalessky, 2023;
- The International Conference on Intelligent Systems and New Applications, Liverpool, 2023;
- The 7th International Scientific Conference of Young Researchers, Baku, 2023;
- The 5th International Conference on Problems of Cybernetics and Informatics (PCI), Baku, 2023;

- The Republican Scientific and Practical Conference entitled "Heydar Aliyev's State Security Policy: Historical Achievements and Contemporary Challenges," Baku, 2025.

**Name of the organisation where the dissertation was carried out.** The dissertation was completed in Department No. 2 of the Institute of Information Technology under the Ministry of Science and Education of the Republic of Azerbaijan.

**Scientific publications.** A total of 23 scientific works have been published on the dissertation topic, including 9 articles in scientific journals and 14 papers in the proceedings of national and international scientific conferences. Of the 9 journal articles, 2 were published in journals indexed in Web of Science, 1 in a journal indexed in Scopus, 1 in a journal indexed in both databases, 2 in journals included in other international indexing systems, and 3 in other scientific journals recommended by the Higher Attestation Commission. Of the 14 conference papers, 3 were published abroad, and 3 were indexed in both Scopus and Web of Science. The author's h-index in the Scopus database is 5, with a total of 76 citations.

**Structure and volume of the dissertation**. The dissertation consists of an introduction, 4 chapters, a conclusion, a list of 169 references, 40 tables, 17 graphs, and 51 figures, with a total length of 175 pages.

## GENERAL CHARACTERISTICS OF THE WORK

The introduction substantiates the relevance of the dissertation topic and defines the aim and objectives of the research. It also presents the scientific novelty and the theoretical and practical significance of the results obtained within the framework of the dissertation.

**In the first chapter,** the types and nature of malware, as well as existing malware analysis methods, were thoroughly investigated to identify current major challenges in the field. Furthermore, previous studies conducted by researchers on malware detection were analyzed. **In the first section of the first chapter,** various types of malware, the operating principles of evasion techniques (metamorphism, polymorphism, and packing), and corresponding countermeasures against these techniques were investigated [18]. **In the second section of the first chapter,** existing malware analysis methods were examined, and their advantages and limitations were analyzed. The techniques and tools used for static and dynamic analysis were presented, and the structure of Portable Executable (PE) files, which constitute the primary object of this research, was investigated [3]. **In the third section of the first chapter, intelligent** analysis methods applied to malware detection were systematically reviewed. The review was conducted in accordance with the PRISMA 2020 methodology, systematically analyzing publications indexed in the Scopus and Web of Science databases to identify the strengths and limitations of existing approaches [3].

The gaps identified as a result of the analysis are presented below:

1. Problems identified in image-based malware detection include the computational burden of fine-tuning pretrained models, their large number of parameters, high resource requirements, and limited feature analysis [4], [11], [20].

2. Based on studies concerning the application of generative adversarial networks (GANs) to malware detection, the following major problems and issues were identified [5], [12], [15]:

- *Data availability***:** The effective training of GANs requires large and diverse datasets. In practice, obtaining malware

samples, particularly zero-day and polymorphic attack samples, is difficult.

- *Class imbalance***:** Malware datasets usually contain more benign samples than malicious ones, making it difficult for GANs to generate samples belonging to rare malicious classes.
- *Resource requirements:* Training GANs requires considerable computational resources and time.
- *Evaluation metrics:* There is a lack of appropriate metrics for measuring the impact and quality of generated malicious samples.

3. The following limitations were identified during the analysis of research on the explainability of malware detection models [17], [22]:

- The reliance of the proposed detection models on datasets consisting only of the same types of malware raises concerns regarding model robustness.
- No instances of testing the proposed detection models in real-time environments were identified in the analysed studies.
- In many studies, malicious and benign files were intentionally distributed equally within the dataset, or the minority samples were duplicated to achieve class balance and artificially improve the accuracy of malware detection models. Although these practices may increase model accuracy, they may also result in a substantial decrease in accuracy in a real testing environment. This issue was not examined in the analysed studies.
- The explainability problem remains unresolved in the majority of existing models. In other words, the features on which the model bases its decisions are not clearly explained. This reduces users' trust in the system, makes it difficult to identify the causes of false-positive and false-negative results, and limits the possibility of deploying such models in real-world environments.

**In the fourth section of the first chapter**, the research tasks related to the development of intelligent analysis methods and algorithms for malware detection were selected from among the

existing problems according to their priority.

**The second chapter** is devoted to the development of a conceptual model for a malware detection system. **In the first section of the second chapter, a** sequential, three-stage conceptual model of the malware detection system, illustrated in Figure 1, was proposed [21]. The first stage encompasses the activities performed prior to the development of the malware detection model. The second stage involves model development, testing, and optimization. At this stage, methods were developed to select the most accurate and robust detection approach, as well as to identify the most relevant features from a large set of available features.

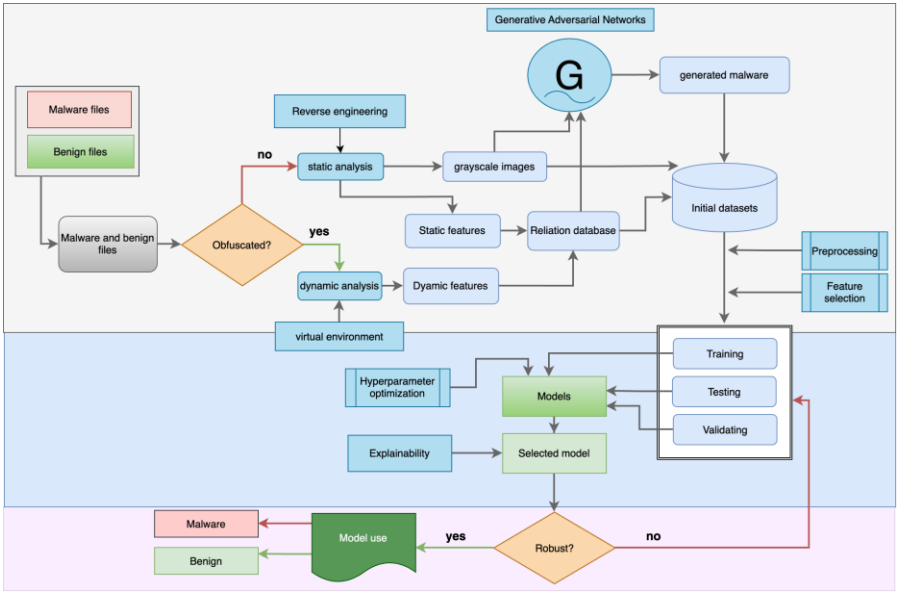



**Figure 1. The proposed conceptual model for malware detection**

The third stage is devoted to demonstrating the explainability of the final method obtained from the second stage, as well as testing and monitoring the robustness of the model in a real-world environment.

As can be seen from the proposed conceptual model, at the first stage, a dataset comprising a specific feature space is formed in preparation for the development of the malware detection model. The features may be obtained through static and dynamic analysis. As a

solution to the problem of selecting the appropriate types of features, a two-stage method was developed in **the second section of the second chapter** for selecting features from executable files, taking into account factors such as the discrete or continuous nature of the features and the linearity of the model [23].

*First stage***.** The first stage is performed independently of the model. At this stage, constant or invariant features, whose variance is equal to zero, relatively constant features, including those with 99% identical values or missing data, and duplicate features are identified and removed. Since these features do not introduce variation into the dataset, they do not contribute to the predictive power of the models. The formula for calculating the variance of continuous features is presented in Formula 1.

$$V(X_i) = \frac{1}{n}\sum_{j=1}^{n}(x_{ij} - \hat{x}_{ij})^2, \quad (1)$$

where $V(X_i)$ denotes the variance of the variable $X_i$, $x_{ij}$ denotes the value of the $j$-th observation of $X_i$, $\hat{x}_{ij}$ denotes the mean value of $X_i$, and $n$ denotes the total number of observations. The obtained variance value is compared with a predefined threshold value.

For discrete features, the percentage of the dominant category is calculated and compared with a predefined threshold to determine whether the feature should be removed. The formula for calculating the percentage of the dominant category for discrete features is presented in Formula 2.

$$D(X_i) = \max\left(\frac{count(c_k)}{n}\right), \forall c_k \in X_i \quad (2)$$

where $D(X_i)$ denotes the percentage of the dominant category, $count(c_k)$ denotes the number of observations belonging to category $c_k$, and $n$ denotes the total number of observations. The obtained percentage of the dominant category is compared with a predefined threshold value.

To remove duplicate continuous features, the Pearson correlation

coefficient presented in Formula 3 is used[4]:

$$r(X_i, X_j) = \frac{\sum(X_i - \bar{X}_\iota)(X_j - \bar{X}_j)}{\sqrt{\sum(X_i - \bar{X}_\iota)^2 \sum(X_j - \bar{X}_j)^2}}, \quad (3)$$

where $r(X_i, X_j)$ denotes the correlation coefficient, while $\bar{X}_\iota$ and $X_j$ denote the mean values of $X_i$ and $X_j$, respectively. The feature selection criterion is determined by comparing the obtained correlation value with a predefined threshold.

To remove duplicate discrete features, the Spearman rank correlation coefficient presented in Formula 4 is used[5]:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}, \quad (4)$$

where $d_i$ denotes the difference between the ranks of the two variables for each observation, and$n$ denotes the total number of observations. If there is no correlation between the ranks, the value of $\rho$ is considered to be equal to 0.

*Second stage***.** At this stage, a decision is made as to whether the remaining features should be retained or removed. During feature selection, the type of model being applied is taken into account, and the selection rule is determined accordingly. In the proposed method, accurate feature selection is performed by considering both the individual effects of features and their interactions. Formula 5 presents the procedure for calculating the individual influence of features:

$$T_i(X_i) = \frac{\sum E(X_i)}{S}, \quad (5)$$

where $T_i(X_i)$ denotes the individual influence of the feature, $E$ denotes entropy, and $S$ denotes the total number of splits.

To account for feature interactions, entropy values are calculated

[4] Ge, G., Zhang, J. Feature selection methods and predictive models in CT lung cancer radiomics // Journal of Applied Clinical Medical Physics, – 2023. 24(1), – Article e13869.

[5] Kim, D., Heo, T.-Y. Anomaly Detection with Feature Extraction Based on Machine Learning Using Hydraulic System IoT Sensor Data // Sensors, – 2022. 22(7), – p. 1-24.

and summed according to the number of splits in which the features are used jointly. Formula 6 presents the procedure for calculating the influence of the joint use of features:

$$T_g(X_i, X_j) = \sum E(X_i, X_j), \quad (6)$$

where $T_g(X_i, X_j)$ denotes the joint influence of the features.

Finally, the general rule that takes into account both the individual and joint influences of the features is presented in Formula 7.

$$T_y(X_i) = \alpha * T_i(X_i) + \beta * \sum_{i \neq j} T_g(X_i, X_j), \quad \alpha + \beta = 1 \quad (7)$$

where $T_y(X_i)$ denotes the final influence score of the feature, while $\alpha$ and $\beta$ are weighting coefficients. The weights are user-defined values ranging from 0 to 1.

Figure 2 presents the general flowchart of the proposed approach to feature selection, in which appropriate methods are applied by considering the discrete or continuous nature of the features at the first stage and the linear or nonlinear nature of the model at the second stage [23].

The proposed malware detection method is aimed at achieving more accurate and effective feature selection, particularly through the use of tree-based models.

**The third section of the second chapter** addresses the development of methods for the explainability of malware detection models and their integration into real-world environments. Taking into account the limitations of existing explainability methods such as SHAP and LIME, a mathematical evaluation method for measuring explanation stability in malware detection models was proposed. The main idea of the proposed approach is that, for samples belonging to the same class and having similar characteristics, the explanations produced by the model should also be statistically similar. If the explanations exhibit high variation, this indicates that the model's decision-making mechanism is unstable and that it reacts excessively and randomly to certain features.

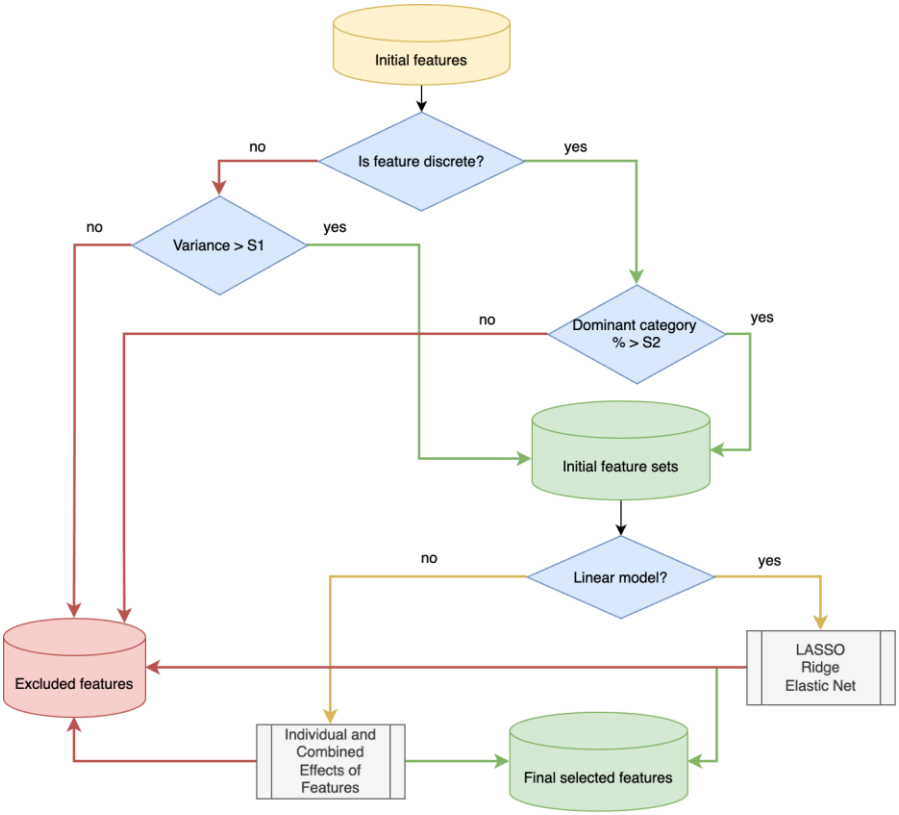



**Figure 2. Block diagram of the proposed approach to feature selection**

For this purpose, samples belonging to the same class are first selected, and the feature attribution vectors defined in Formula 8 are calculated for each sample using SHAP or LIME:

$$\Phi^{(k)} = \left(\phi_1^{(k)}, \phi_2^{(k)}, \dots, \phi_n^{(k)}\right), k = 1,2, \dots, K, \qquad (8)$$

where $n$ denotes the number of features used in the model, and $\phi_i^{(k)}$ denotes the explanatory impact value of the $i$-th feature for the $k$-th sample. The variability of feature attributions is determined for each feature using Formula 9:

$$\mathrm{Var}_i = \mathrm{Var}\left(\phi_i^{(1)}, \phi_i^{(2)}, \dots, \phi_i^{(K)}\right) \quad (9)$$

Based on this variability, the explanation stability indicator for a feature is proposed according to Formula 10:

$$\mathrm{Stab_i} = 1 - \frac{\mathrm{Var}_i}{\max\limits_{j=1,\dots,n} \mathrm{Var}_j}, \quad (10)$$

where $\mathrm{Stab}_i \in [0,1]$ and expresses the degree of stability of the explanation of the $i$-th feature. A value close to 1 indicates that the contribution of the feature to the model's decision is stable across samples, whereas a value close to 0 indicates that the explanation is unstable in nature.

The overall explanation stability of the model is determined as follows:

$$\mathrm{Stab}(f) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Stab}_i \quad (11)$$

This indicator characterises the overall reliability level of the model in terms of explainability and may be used for the comparative analysis of the explainability quality of different detection models, such as Random Forest, XGBoost, CNN, and LSTM.

**The third section of the second chapter** examines various issues related to assessing the robustness of detection models during their operation in real-world environments [14]. The first issue involves the use of the Population Stability Index (PSI) method to identify changes in data, while the second concerns monitoring deviations in model performance. The proposed model incorporates adaptive monitoring mechanisms based on PSI and performance metrics. Thus, the system to be developed will not only demonstrate high accuracy under laboratory conditions but will also meet modern cybersecurity requirements in terms of sustainability, reliability, and transparency in real-world environments. The practical results of the proposed method are discussed in the fourth chapter.

**In the third chapter,** new models were developed for malware encountered in local environments running Windows and Android operating systems through the application of binary and multiclass classification techniques, clustering methods, image-based classification approaches, Generative Adversarial Network based

techniques, and autoencoder methods. Experimental studies were conducted for software running on both Windows and Android operating systems in accordance with the first and second stages of the conceptual model proposed in the second chapter. These studies involved the application of binary and multiclass classification methods using tabular malware data, clustering methods, image-based classification methods, generative adversarial network-based methods, and an autoencoder approach. **The first section of the third chapter** examines the development of machine learning methods, including binary classification, multiclass classification, and clustering, for malware detection [1], [2], [6-8], [10], [13], [19]. In the first of these studies, a highly accurate machine learning model was developed using opcodes, namely operation codes extracted from malware samples [13]. The results showed that although tree-based algorithms achieved relatively high performance on the dataset constructed from opcode frequencies, certain limitations remained. In particular, feature-importance analysis revealed that frequently occurring opcodes such as “push” and “move” demonstrated similar levels of importance in both malicious and benign software, thereby raising doubts about their discriminative capability. Consequently, it became necessary to investigate other types of features [13]. In the second study, a highly accurate machine learning model was developed for the specific task of detecting malware traffic using the MTA-KDD19 malware traffic dataset [7]. With regard to the Android operating system, which constitutes one of the principal research objects, this section considers not only binary malware detection but also multiclass malware classification using the CICMalDroid and Drebin datasets [6], [8], [10]. The proposed general framework is illustrated in Figure 3.

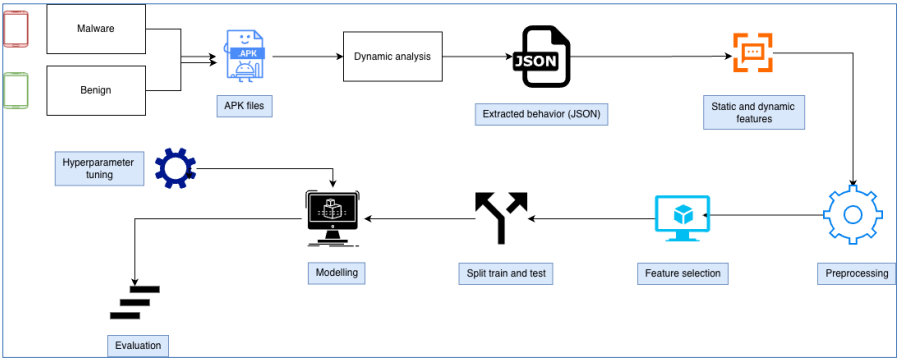



**Figure 3. Proposed scheme for detecting malicious mobile applications**

To eliminate the labelling problem associated with malware samples and to determine, in groups, whether suspicious samples are malicious or benign, an unsupervised clustering method that does not require a labelling process was also developed [2]. The K-modes model achieved high performance when (k=2). The following conclusions were drawn from the study:

- Signature-based methods cannot detect new malware or malware with dynamically changing signatures.
- Although dynamic analysis is highly effective, its real-time application in large-scale systems is not feasible.
- The development of hybrid approaches presents considerable difficulties and frequently results in interoperability problems between systems.
- The increasing complexity of malware and the use of encryption and obfuscation techniques to conceal malicious behaviour reduce the effectiveness of traditional detection methods [2].

**The second section of the third chapter** addresses the development of a deep learning-based malware detection method. Although high performance can be achieved using existing datasets, the dependence of model performance on feature quality and the limitations associated with detecting new types of malware demonstrate the scientific and practical necessity of developing and empirically evaluating methods that provide more effective data representation, particularly image-based detection methods. The two

feature families most frequently used in behavioural analysis are system calls and API calls. Although deep learning models have demonstrated promising results in malware identification and classification based on these features, their robustness against malware designed to evade detection, as well as their high computational and memory requirements, must also be considered [9].

Two approaches were examined in this context. The first approach involved converting malware samples into images and classifying them using deep learning methods [4], [11], [20]. The second approach involved applying an autoencoder to malware images and reformulating malware detection as an anomaly-detection problem [16]. The main principle of image-based malware detection is that PE files are first converted into binary code, after which the resulting code is transformed into a standard format suitable for image generation. Standard-sized greyscale images are then obtained using various transformation techniques [16].

Thirteen transfer learning methods were evaluated on the Malimg dataset, and a highly accurate and robust model was developed using MobileNet. Figure 4 illustrates the changes in the numerical values of accuracy and loss for the training and validation datasets as the number of epochs increases.

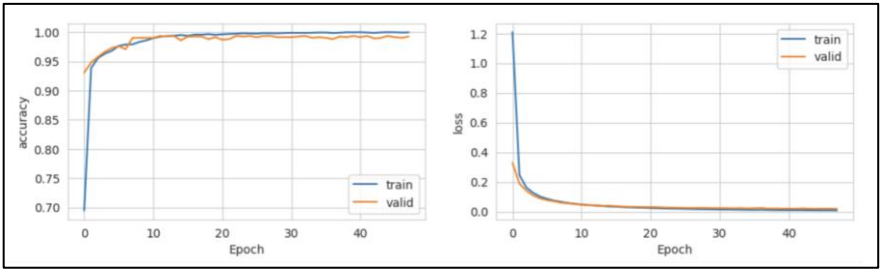



**Figure 4. MobileNet results**

The graph clearly demonstrates that the results obtained for the training and validation datasets are both high and differ only slightly from one another. This indicates that the model exhibits a high level of robustness. Table 1 compares the results obtained using the proposed method with those reported by other authors using the same dataset.

**Table 1. Comparative analysis of malware detection methods evaluated on the Malimg dataset**

| Author | Year | Proposed method | F1-score (%) |
|---|---|---|---|
| Hai T. et al. | 2023 | Inception V3[6] | 74.29 |
| Kim J. et al. | 2023 | Attention-based cross-modal CNN[7] | 97.6 |
| Ahmed I.T. et al | 2024 | Segmentation-based fractal texture analysis and KNN[8] | 70.14 |
| **Proposed method** | **2024** | **MobileNet** | **98.9** |

Despite the high performance achieved through image-based methods, several problems remain, including the sensitivity of the results to class imbalance, data diversity, and distributional changes. The conversion of binary files into images may result in the loss of certain semantic signals contained in the original files, weaken generalisation for underrepresented classes, and reduce robustness against newly developed evasion techniques. Furthermore, the computational cost of classification and its dependence on class quality impose limitations on practical applications.

For this reason, malware detection was formulated as an anomaly-detection problem, and autoencoders were employed [16]. In the proposed approach, the input data are represented either as greyscale malware images or as executable-feature vectors. An encoder–decoder architecture, such as a convolutional autoencoder, is trained using reconstruction losses or structural similarity-based loss

[6] Hai, T.H. A Proposed New Endpoint Detection and Response With Image-Based Malware Detection System / T.H. Hai, V. Van Thieu, T.T. Duong [et al.] // IEEE Access, – 2023. 11, – p. 122859–122875.

[7] Kim, J., Paik, J.Y., Cho, E.S. Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification // IEEE Access, – 2023. 11, – p. 22889–22903.

[8] Ahmed, I.T., Hammad, B.T., Jamil, N. A Comparative Performance Analysis of Malware Detection Algorithms Based on Various Texture Features and Classifiers // IEEE Access, – 2024. 12, – p. 11500–11519.

functions. The detection threshold is calibrated on the validation dataset using ROC analysis or by maximising the F1-score.

This approach offers two main advantages: (i) it reduces dependence on predefined classes and increases sensitivity to new, previously unseen, or zero-day malware families; and (ii) it can complement existing supervised image-based models within a hybrid framework by functioning as a preliminary filtering or signal-enhancement stage.

Several threshold values were tested to determine the optimal threshold that maximised classification performance. Following multiple experiments, a threshold value of 0.95 was found to produce the best results. Thus, when the reconstruction error of an input image is below 0.95, the sample is classified as malware, whereas an error above 0.95 results in its classification as benign.

The model was evaluated on the test dataset, and a confusion matrix and various performance metrics, including accuracy, precision, recall, and F1-score, were calculated. According to the confusion matrix, the model achieved an overall accuracy of 62.3%. Precision, which measures the proportion of samples predicted as malicious that were actually malicious, was 58.1%. Recall, which reflects the model's ability to identify actual malicious samples, reached a high value of 91.9% [16]. Recall is particularly important in malware detection tasks because failing to identify malicious samples may lead to serious security risks. **In the third section of the third chapter**, a conditional generative adversarial network-based model was proposed to address the imbalance in the dataset used to develop the malware detection model [15]. In conditional GANs, the generation process depends not only on random noise (z), but is also guided by an additional condition (y), such as a class label, an attribute, or other supplementary information. This approach allows the generator to create samples corresponding to the specified condition, while the discriminator evaluates both whether the sample is real or synthetic and whether it is consistent with the given condition.

Consequently, conditional GANs are widely applied to controlled generation problems, including the mitigation of class imbalance,

image-to-image translation, and attribute-conditioned image synthesis[9]. The objective function of the conditional GAN is presented in Formula 12 [15]:

$$min_G max_D V(D,G) = E_{(x,y)\sim p_{data}(x,y)}[logD(x|y)] + \\ + E_{z\sim p_z(z),y\sim p(y)}\big[log\big(1 - D(G(z|y)|y)\big)\big] \qquad (12)$$

where $y$ denotes the condition or label, such as a class or attribute; $D(x|y)$ denotes the discriminator, which evaluates whether a sample is real and whether it is consistent with the specified condition; $G(z|y)$ denotes the generator, which produces synthetic samples corresponding to the condition; $p_{data}(x,y)$ denotes the joint distribution of the real data; and $p_z(z)$ denotes the generator's input noise, such as a normal or binomial distribution. Table 2 presents a comparison of the models evaluated on the original dataset before synthetic sample generation, whereas Table 3 presents a comparison of the models evaluated on the balanced dataset after the application of the conditional GAN.

**Table 2. Comparison of Models on the Imbalanced Dataset**

| Models | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| LightGBM | 0.9153 | 0.8958 | 0.8643 | 0.8798 |
| Random Forest | 0.9189 | 0.9096 | 0.8593 | 0.8837 |
| KNN | 0.8559 | 0.8083 | 0.7839 | 0.7959 |
| Logistic Regression | 0.7532 | 0.7460 | 0.4724 | 0.5785 |

**Table 3. Comparison of Models on the Balanced Dataset**

| Models | Accuracy | Precision | Recall | F1-measure |
|---|---|---|---|---|
| ***LightGBM*** | ***0.9568*** | ***0.9619*** | ***0.9541*** | ***0.9579*** |
| Random Forest | 0.9513 | 0.9666 | 0.9378 | 0.9520 |
| KNN | 0.8955 | 0.9041 | 0.8919 | 0.8980 |
| Logistic Regression | 0.8078 | 0.8816 | 0.7243 | 0.7953 |

[9] Imamverdiyev, Y.N., Musayeva, F.I. Analysis of generative adversarial networks // Problems of Information Technology, – Baku: – 2022, – 13 (1), – p. 20–27.

As can be seen from Tables 2 and 3, higher performance was achieved on the balanced dataset. These results demonstrate that conditional GAN-based data generation is an effective approach for reducing class imbalance, improving the precision–recall balance of the model, and ensuring more stable overall performance in malware detection.

**In the fourth chapter,** the interpretability of malware detection model decisions and their robustness under temporally evolving data conditions were evaluated using various publicly available datasets. The obtained results contribute to assessing the applicability of these models in real-world operational environments. The practical reliability and robustness of the developed malware detection model in real-world environments are substantiated through two complementary approaches:

(i) model explainability [17], [19], [22];

(ii) systematic monitoring of robustness in real-world environments [14].

The primary objective is to demonstrate that the selected final model is capable of producing reliable, sustainable, and practically applicable results in real-world environments. The deployment of the final model is intended to ensure not only its successful performance under theoretical and experimental conditions, but also its effectiveness in practical operating environments. **The first section of the fourth chapter** provides a practical evaluation of the explainability of malware detection models in real-world environments using three different datasets [17], [19], [22]. For this purpose, various machine learning models were first developed using different datasets and compared on the basis of several performance metrics, while also taking model training time into account.

The overall results of the experiments demonstrate that, among the models evaluated on the "CICMalDroid2020," "CICMalMem2022," and "Irfan Yousuf" datasets, LightGBM emerged as the most stable and high-performing approach in both multiclass and binary classification scenarios:

- "CICMalDroid2020": F1 = 0.93 and ROC-AUC = 0.99 [22];
- "CICMalMem2022": F1 = 0.81 and ROC-AUC = 0.95 [19];

- “Irfan Yousuf”: F1 = 0.93 and ROC-AUC = 0.99 [17].

From the perspective of evaluating explainability in real-world environments, SHAP and LIME analyses clearly revealed the logical basis of the model’s decisions. In the Android environment, the principal features influencing class-specific decisions included access to device identifiers and personal information, such as getDeviceId, getSubscriberId, and ACCESS_PERSONAL_INFO; network activity, such as NETWORK_ACCESS; accessibility and user-interface manipulation, such as sendAccessibilityEvent; and file and system calls, such as pread64.

For the memory-based CICMalMem2022 dataset, process-, service-, and processing-related features played an important role in distinguishing between classes. In the Windows PE environment, structural indicators associated with headers, sections, DLLs, and APIs, including TimeDateStamp, DLLCharacteristics, and CheckSum, played a major role in differentiating malware samples.

Overall, the results confirm that LightGBM is a more appropriate and reliable model for deployment in real-world environments, not only because of its high detection accuracy but also because its decisions can be explained using SHAP and LIME.

**The second section of the fourth chapter** provides a practical evaluation of the robustness of malware detection models in real-world environments [14]. For the experiment, the Drebin dataset, consisting of malware samples recorded between August 2010 and October 2012, was used to assess model robustness across different time intervals using the proposed methods.

To develop the model, data from the first two years of the dataset were initially used, while the data from the final year were retained for robustness evaluation. Various algorithms, including logistic regression and several tree-based algorithms, were used for model training. LightGBM demonstrated the highest performance and robustness among the evaluated methods, achieving an F1-score of 0.98.

At the next stage, the training process was repeated using the data from the final year reserved for the experiment in order to assess the stability of the LightGBM algorithm. The feature-importance scores

of the two models were compared, and no significant differences were observed between the importance rankings of the features.

In addition, to evaluate population stability, the datasets used in the first and second experiments were assessed using the PSI method. Table 4 presents only the features whose drift values were equal to or greater than 0.01.

**Table 4. PSI values**

| Features | Drift value |
|---|---|
| Landroid.content.Context.unregisterReceiver | 0.02 |
| Ljava.lang.Class.getResource | 0.01 |
| Ljava.net.URLDecoder | 0.01 |
| WRITE_HISTORY_BOOKMARKS | 0.01 |
| transact | 0.01 |
| TelephonyManager.getLine1Number | 0.01 |
| SEND_SMS | 0.01 |
| onServiceConnected | 0.01 |
| Ljava.lang.Class.getMethods | 0.01 |
| RECEIVE_SMS | 0.01 |

As can be seen from the table, minor changes in the dataset do not have a significant effect on model performance, and the model remains stable. The comparative experiments and drift monitoring demonstrate that the final solution is robust to newly arriving data and minimises the risk of incorrect decisions.

The experiments demonstrate similarities in feature importance and minor shifts in data distributions across the selected time intervals, supporting the applicability of the monitoring approach. However, confirming the model's robustness over time requires evaluating it without retraining on independent temporal test sets and under real operating conditions using precision, recall, and F1-score.

## MAIN RESULTS OF THE DISSERTATION

During the entire research conducted within the scope of the dissertation topic, the scientific tasks were addressed by the author, and the following main results were obtained:

1. Existing international practices in the field of malware detection were systematically analysed, and the principal development trends, technologies applied, current methodological limitations, and research gaps in the field were identified. Bibliometric and comparative analyses based on quantitative and qualitative indicators were conducted for publications indexed in various scientific databases [3], [9], [12], [16], [18]. The research findings can be used by cybersecurity organizations to select modern detection approaches and define development strategies.

2. A three-stage conceptual model was developed to ensure the robust and effective detection of malware. In addition to covering the stages of malware sample preprocessing, detection model development, and deployment in real-world operating environments, the proposed model provides for the evaluation of the explainability of model decisions, the monitoring of performance indicators, and the assessment of model robustness [21]. Applying the three-stage conceptual model in Security Operations Centers can help automate preprocessing, detection, decision explanation, and monitoring processes.

3. A hybrid filtering method that takes into account the discrete and continuous nature of features was developed for selecting informative features from the static and dynamic feature space used in the development of machine learning models. The proposed method enabled the dimensionality of the feature space and computational complexity to be reduced while improving the efficiency of the detection process [23]. The hybrid feature selection method is recommended for reducing computational overhead and enabling the rapid processing of large datasets.

4. Classification-based and clustering-based machine learning methods that use various types of features obtained through static and dynamic analysis were developed for detecting malware and malicious network traffic in Windows and Android operating system

environments. The effectiveness of the proposed methods in binary and multiclass classification tasks and in clustering similar samples was evaluated experimentally [1], [2], [6-8], [10], [13]. Both classification and clustering methods can be used for the timely detection of malware and malicious traffic on personal computers, Android devices, and networks.

5. A highly accurate, deep transfer learning-based algorithm was successfully developed for detecting malware by efficiently converting raw malware samples into visual image representations. Comprehensive experimental results demonstrated that the fine-tuned MobileNet model achieved exceptionally high and robust classification performance, reaching an outstanding F1-score of 98.9% [4], [11], [20]. Consequently, this lightweight MobileNet-based approach can be effectively applied to real-time image-based analysis of malicious code on various resource-constrained mobile devices.

6. A method for generating synthetic samples based on a conditional generative adversarial network was developed to increase the number of malware samples in the training data and reduce class imbalance. Machine learning models were evaluated on the dataset augmented with the generated samples, and improvements in their detection performance were observed [5], [12], [15]. Generating synthetic samples using a conditional generative adversarial network can increase the representation of rare malware in training datasets and reduce class imbalance.

7. Comprehensive experimental studies across three diverse benchmark datasets rigorously evaluated the explainability of the proposed malware detection models using SHAP, LIME, and feature-importance analysis [17], [19], [22]. Additionally, the Population Stability Index (PSI) was thoroughly evaluated to monitor longitudinal changes in underlying data distributions [14]. Together, these robust analytical tools empower security analysts to interpret model predictions with high confidence, proactively assess the critical need for timely model updates, and dynamically select appropriate protective countermeasures against emerging threats.

**The main results of the dissertation have been published in the following scientific works:**


1. Bağırov, E.O. Zərərli proqramların n-qram analizi əsasında aşkarlanması // Tələbə və gənc tədqiqatçıların I beynəlxalq elmi konfransının materialları, – Bakı, – 6–16 aprel, – 2020, – s. 431–433.
2. Bağırov, E.O. Klasterizasiya üsullarının zərərli proqramların aşkarlanmasına tətbiqi // Tələbə və gənc tədqiqatçıların II beynəlxalq elmi konfransının materialları, – Bakı, – 13–28 aprel, – 2021, – s. 438–440.
3. Baghirov, E. Techniques of Malware Detection: Research Review // 15th International Conference on Application of Information and Communication Technologies (AICT), – Baku: IEEE, – 13–15 October, – 2021, – p. 1–6. **(Scopus, WoS)**
4. Baghirov, E. Image-Based Malware Detection Method // Proceedings of the II International Conference on Information Security: Problems and Prospects, – Baku: Azerbaijan University, – 25 November, – 2022, – p. 28–31.
5. Bağırov, E.O. Zərərli proqramların generativ rəqib şəbəkələrlə generasiyası // "Rəqəmsal Transformasiya–2023" IV beynəlxalq elmi konfransının materialları, – Bakı, – 18–19 aprel, – 2023, – s. 378–380.
6. Baghirov, E. Machine Learning-based Android malware detection: a robust and accurate approach // Proceedings of the VII International Scientific Conference of Young Researchers, – Baku, – 28–29 April, – 2023, – p. 1–10.
7. Baghirov, E. Leveraging Machine Learning for Accurate Malware Traffic Detection // Proceedings of the International Conference on Intelligent Systems and New Applications (ICISNA'23), – Liverpool, – 28–30 April, – 2023, – p. 25–26.
8. Baghirov, E. Evaluating the Performance of Different Machine Learning Algorithms for Android Malware Detection // Proceedings of the 2023 5th International Conference on Problems of Cybernetics and Informatics (PCI), – Baku: IEEE, – 28–30 August, – 2023, – p. 1–4. **(Scopus, WoS)**

9. Baghirov, E.O. Analyzing the performance of behavioral-based malware detection approaches under real-world conditions // Optical-Electronic Devices and Devices in Pattern Recognition and Image Processing Systems: Proceedings of the XVII International Scientific and Technical Conference, – Kursk: South-West State University, – 12–15 September, – 2023, – p. 15–17. (**РИНЦ)**
10. Baghirov, E. Comprehensive Framework for Malware Detection: Leveraging Ensemble Methods, Feature Selection and Hyperparameter Optimization // Proceedings of the 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), – Baku: IEEE, – 18–20 October, – 2023, – p. 1–5. **(Scopus, WoS)**
11. Baghirov, E. Disassembled files to grayscale images: Advancing malware detection with diverse features // Doktorantların və gənc tədqiqatçıların XXVI respublika elmi konfransının materialları, – Bakı, – 10–11 noyabr, – 2023. Cild 1, – p. 130–134.
12. Baghirov, E.O. Critical analysis and review of current research on GANs for malware detection // Proceedings of the 12th National Supercomputer Forum, – Pereslavl-Zalessky, – 28 November–1 December, – 2023, – p. 1–10.
13. Baghirov, E. Malware detection based on opcode frequency // Problems of Information Technology, – 2023. 14(1), – p. 3–7.
14. Baghirov, E. A comprehensive framework for real-time malware detection and monitoring in production // International Journal on Information Technologies and Security, – 2024, – 16(4), – p. 85-94. **(WoS)**
15. Baghirov, E. Building robust malware detection through Conditional Generative Adversarial Network-based data augmentation // Program Systems: Theory and Applications, – 2024. 15(4), – p. 97–110. (**WoS**, **РИНЦ)**
16. Baghirov, E., Afzal, M. Unveiling Malicious Patterns: Autoencoder-Based Malware Detection // System Informatics, – 2024. 24, – p. 113–122. **(РИНЦ)**

17. Baghirov, E. Advanced machine learning and interpretability for Windows malware detection // Journal of Modern Technology and Engineering, – 2024. 9(3), – p. 165–177.
18. Imamverdiyev, Y.N., Baghirov, E.O. Evasion Techniques in Malware Detection: Challenges and Countermeasures // Problems of Information Technology, – 2024. 15(2), – p. 9–15.
19. Imamverdiyev, Y., Baghirov, E., Ikechukwu, J. Detecting obfuscated malware infections on Windows using ensemble learning techniques // Informatics and Automation, – 2025. 24(1), – p. 99–124. **(Scopus, WoS)**
20. Imamverdiyev, Y., Baghirov, E., Ikechukwu, J.C. Image-based Deep Learning method for effective malware detection // Problems of Artificial Intelligence, – 2025. № 1(36), – p. 106–122. **(РИНЦ)**
21. Bağırov, E.O. Zərərli proqramların aşkarlanması sisteminin ümumi strukturu // "Mütərəqqi texnologiyalar və innovasiyalar" X respublika elmi-texniki konfransının materialları, – Bakı, – 1–2 may, – 2025, – s. 310–313.
22. Baghirov, E. A comprehensive investigation into robust malware detection with explainable AI // Cyber Security and Applications, – 2025. 3, – Article 100072. **(Scopus – Q1)**
23. Bağırov, E.O. Zərərli proqramların aşkarlanması modelləri üçün əlamətlərin seçilməsi üsulu // "Heydər Əliyevin dövlət təhlükəsizliyi siyasəti: tarixi nailiyyətlər və müasir çağırışlar" respublika elmi-praktik konfransının materialları, – Bakı, – 6 may, – 2025, – s. 294–299.

**The applicant's personal contribution to the co-authored publications:**

[16] – Dataset preparation; methodology; visualisation; interpretation of the results.

[18] – Literature review; writing of the manuscript.

[19] – Methodology; experiments; visualisation; interpretation of the results; writing of the manuscript.

[20] – Methodology; experiments; visualisation; interpretation of the results; writing of the manuscript.

The defense of the dissertation will be held on 27 November 2026 at $16^{00}$ at the meeting of the FD 1.35 Dissertation Council operating under the Institute of Information Technology of the Ministry of Science and Education of the Republic of Azerbaijan.

Address: Az 1141, Baku city, B.Vahabzadeh street, 9a.

The dissertation can be viewed in the library of the Institute of Information Technology of the Ministry of Science and Education of the Republic of Azerbaijan.

Electronic versions of the dissertation and abstract are posted on the official website of the Institute of Information Technology of the Ministry of Science and Education of the Republic of Azerbaijan.

The abstract was sent to the necessary addresses on 26 October 2026.

Signed for print: 22.10.2026

Paper format: A5

Volume: 39885 characters

Number of hard copies: 20