

HYBRID NETWORK INTRUSION DETECTION SYSTEM BASED ON SIGNATURE ANALYSIS AND MACHINE LEARNING METHODS

IGISSINOV S.ZH.¹, RYSKELDIKYZY R.¹, GAISAGALEYEVA B.M.²,
UVALIYEVA G.M.³, ZHUMABAYEVA ZH.O.^{4*}

Igissinov Sabit Zhandarbekovich¹ – PhD, acting associate professor, department of mathematics, M.Kh. Dulaty Taraz university, Taraz, Kazakhstan,

E-mail: igissinovsabit@gmail.com, <https://orcid.org/0000-0002-4189-9142>

Ryskeldikyzy Rabiga¹ – Master, senior lecturer, department of mathematics, M.Kh. Dulaty Taraz university, Taraz, Kazakhstan

E-mail: makulbekova.1985@mail.ru, <https://orcid.org/0009-0003-0076-203X>

Gaisagaleeva Bakhit Mukhitovna² – Master, computer science teacher, Aktobe Nazarbayev intellectual school of natural sciences and mathematics, Aktobe, Kazakhstan

E-mail: gaisagaleeva_m@akb.nis.edu.kz, <https://orcid.org/0009-0001-2419-0414>

Uvaliyeva Gulsiya Maksatovna³ – 2nd year master's student, department of computer technologies and artificial intelligence, Sh. Yessenov Caspian state university of technology and engineering, Aktau, Kazakhstan.

E-mail: w_g_w@mail.ru, <https://orcid.org/0009-0008-6151-0457>.

***Zhumabayeva Zhanar Orynbasarovna**⁴ – Master of technical sciences, M. Auezov South Kazakhstan research university, Shymkent, Kazakhstan.

E-mail: janaru_77@bk.ru, <https://orcid.org/0009-0007-8906-1801>

Abstract. In the context of increasing complexity of modern network infrastructures and the growing number of cyberattacks, improving the effectiveness of intrusion detection systems is a highly relevant task. The aim of this work is to develop and investigate a hybrid approach to network attack detection based on a combination of signature-based analysis and intelligent machine learning methods. The research methods include analysis of requirements for anomaly detection systems, comparative analysis of existing solutions, design of a hybrid system architecture, and experimental modeling using multiclass classification of network traffic. The results demonstrate that integrating a neural model with a traditional detection system improves attack detection coverage and reduces the number of false positives. The practical value is represented by the ability to use the developed scheme on actual enterprise network infrastructure, theoretical - proof of efficiency of mixed method for analysis of traffic. Further studies will be focused on learning models adaptively and increasing number of identified attacks.

Key words: information security, intrusion detection, network traffic, anomalies, machine learning, neural networks, attack classification.

Introduction

Contemporary computer networks have increased complexity, variable loads and an ever-growing amount of different attacks. Growth of network capacity, use of cloud and distributed resources, new kinds of attacks make it hard to maintain information security. Under such conditions intrusion detection systems are essential for securing the network resources through surveillance of traffic and alerting on time to any suspicious behavior.

Conventional intrusion detection using signatures can identify well-known attacks quite well but falls short when encountering novel, altered or unseen threats. At the same time, machine-learning techniques and neural networks allow for identification of network traffic anomalies based on statistical and behavioral analysis of traffic, however usually have large number of false positives and a significant dependency on training data quality. This leads to the need for developing hybrid approaches combining signature-based and mining methods.

The hypothesis of the study is that the integration of signature analysis and machine learning methods within the framework of a hybrid intrusion detection system will increase the completeness of detecting network attacks and reduce the number of false positives compared with the separate use of these approaches.

The purpose of this work is to develop and research a hybrid network attack detection system that combines signature analysis and machine learning methods to improve the detection of anomalies

and attacks in network traffic.

Thus, the research is aimed at improving hybrid methods for detecting network attacks and expanding the possibilities of practical application of intelligent computer network security systems.

Traditional intrusion detection systems based on signature analysis are widely used in practice due to the high speed of network traffic processing and the accuracy of detecting known attacks [1; 2]. However, such systems are significantly limited by their dependence on predefined rules and signatures, which reduces their effectiveness in detecting new and modified attacks, including zero-day threats [3].

Review studies of modern IDS emphasize that the most promising direction for their further development is the transition to systems based on anomaly detection, as well as to hybrid systems combining signature analysis with machine learning methods [1; 4]. Such systems increase the completeness of attack detection and reduce the number of false positives by intelligently analyzing the behavioral characteristics of network traffic [5]. At the same time, there are problems related to scalability, class imbalance, and model stability in real network environments [4; 6].

A significant amount of research is devoted to the application of classical machine learning algorithms to detect network anomalies. Such studies use the method of support vectors, random forests, gradient boosting methods, and dimensionality reduction methods [7; 8]. Experimental results demonstrate high classification accuracy, but emphasize the need for careful selection of features and the limited adaptability of such models to changing network operating conditions [7].

With the development of deep learning, more and more attention is being paid to neural networks that can automatically identify complex patterns in network traffic. A number of studies have proposed convolutional, recurrent, and hybrid architectures, as well as models based on attention mechanisms for analyzing the spatiotemporal characteristics of traffic [9-11]. Experiments show an increase in the completeness and accuracy of detection, including when detecting distributed and hidden attacks; however, such solutions require significant computing resources and large amounts of labeled data [10; 11].

A separate area of research is related to multilevel and two-stage intrusion detection systems, in which the binary classification («norm/attack») is complemented by a multiclass identification of attack types [12]. This approach allows not only to detect the fact of an invasion, but also to determine its nature, which is important for choosing appropriate response measures. At the same time, the complexity of multiclass classification is emphasized in conditions of class imbalance and similarity of signs of various types of attacks [5; 12].

A number of methodological studies draw attention to the limitations of the use of machine learning in network security tasks, including data set bias, markup inaccuracies, and overestimation of the effectiveness of models in the laboratory [6]. These studies conclude that it is advisable to integrate machine learning models with traditional IDS, in which intelligent methods act as an additional analysis mechanism rather than a complete replacement for signature tools [3; 6].

Based on a review of the literature, hybrid systems for intrusion detection that integrate signature-based methods with machine learning techniques show the most promise [1; 4; 13]. Nonetheless, while significant advances have been achieved, there are a number of barriers to be overcome, such as decreasing the number of false positives, adapting the methods for each particular network environment, and ensuring stable operation of the IDS under extreme load. This indicates the necessity for further investigations.

Materials and methods of research

In order to ensure proper work of the intelligent anomaly detection system in the modern network environment, where complexity increases and a growing number of cyber threats arise, it should fulfill several technical requirements. These requirements determine the effectiveness, adaptability, and scalability of the system (Figure 1).

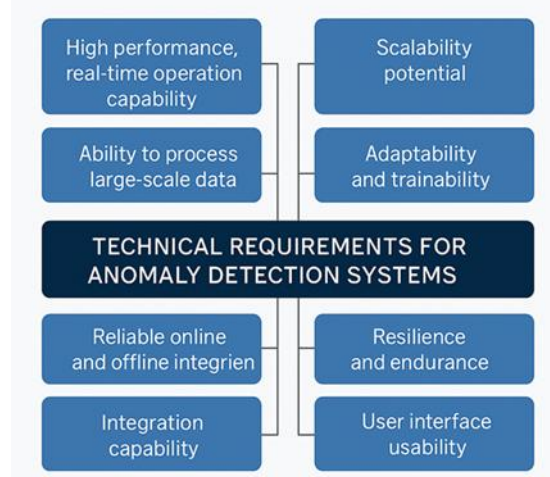


Figure 1. Structural diagram of technical requirements for an anomaly detection system

The reliability and effectiveness of the anomaly detection system mostly depend on its capacity to process a considerable volume of information, to analyze network traffic in real time, to accommodate various types of traffic, to scale with respect to the increasing load, and to cooperate with other security systems. These requirements create the foundation for the architecture and Artificial neural networks (ANNs) have numerous benefits as compared to traditional signature-based approaches for anomaly detection. Since ANNs can approximate complex functions, they can learn complicated attack patterns which are difficult to specify manually. Hence, the network creates its own representation of distinguishing features which help spot normal behavior and malicious activity. Previous research has shown that deep learning models are able to uncover sophisticated attacks that traditional intrusion detection systems could fail to notice [14].

Nonetheless, ANN-based approaches have some disadvantages as well. Their decisions may be hard to interpret; their results are dependent to a large extent on the quality and representativeness of the data used to train the model and they may produce inaccurate classifications for unseen patterns. Such problems are being gradually solved by creating methods of explainable artificial intelligence (XAI) and combining the use of neural networks with signature-based methods. The combination of these two approaches provides a viable solution for uncertain and unfamiliar events. For instance, if all classes have low softmax probabilities, the event can be categorized as «unknown» and send for further inspection.

The experimental findings from this study highlight the benefits of merging signature-based detection with neural network processing. Suricata detects almost all previously known attacks accurately and quickly, while the ANN identifies unusual events that might not be related to existing signatures. By placing these events in their broader context, the neural network also contributes to the minimization of false positives. In general, the hybrid method formed by combining Suricata with an ANN provides more successful results in terms of detection rate than either of the two methods used alone.

The advantages of ANNs over purely signature-based analysis include broader threat coverage (including new attack types), consideration of multicomponent features, the ability to learn from data, adaptability, and the potential reduction of both missed attacks and false positives [15; 16]. These advantages make artificial intelligence methods a promising direction for the development of information security tools, which is also confirmed by the results of the research prototype developed in this work.

For the practical implementation of AI-based anomaly detection, an integrated system consisting of several interrelated components was designed. The overall system architecture is shown in Figure 7 and includes the following main modules:

Traffic Capture Module (IDS Sensor) — responsible for real-time network traffic interception and its initial analysis. In the proposed implementation, this role is performed by Suricata, a high-

performance IDS/IPS. Suricata is connected to traffic monitoring at the network perimeter node (for example, via a switch SPAN port or in gateway mode). It analyzes each passing packet by matching it against a rule set and stores detailed information about sessions and events. Suricata is configured in IDS mode (detection only, without blocking) and outputs logs in JSON format (Eve log). These logs contain signature-based alerts, network flow characteristics (flow records), as well as additional data such as HTTP requests, DNS, and TLS handshakes. Thus, Suricata serves as a data source both for classical signature-based detection (its own alerts) and for subsequent neural network analysis.

Data Processing Module – an intermediate component that receives raw logs from Suricata and transforms them into a format suitable for input to the AI model. In simple terms, it extracts features from network events. The Python programming language plays an important role in data processing in this system by utilizing two libraries: Pandas and Scikit-learn. The selection of pertinent events occurs in this phase as only completed sessions or particular protocol types are taken into consideration, depending on the requirements, followed by the grouping of the data according to the sessions and feature computation. For every TCP session, the extraction of information from Suricata's logs is done for the extracted data, including the volume of transferred bytes and packets in both directions, TCP command flags, time duration of the session, the average speed of transferring data, port information, and any Suricata alerts based on the signature check. After that, all the attributes are scaled according to the training set, either using min/max values or by normalizing them using mean and standard deviation. The result is a vector of features for each session ready to be introduced into the neural network. The processing subsystem operates almost in real-time as it checks the Suricata log file eve.json and does the processing immediately.

Artificial Neural Network (ANN) Model – The Artificial Neural Network (ANN) model is at the core of the framework. It's the trained neural network that resides in memory to predict on data in real time. The neural network is made with TensorFlow/Keras. During the process of the framework initialization, the model is loaded from the saved .h5 file and the session feature vectors are fed to it. The output is a probability distribution over classes (softmax output), which the data processing module interprets as a specific class (for example, «normal traffic» or a particular type of attack). For consistency and convenience, the model classes were mapped to categories used in Suricata (although Suricata does not perform attack classification directly, indirect identification is possible; for example, the presence of a DDoS signature corresponds to the “DDoS” class). The ANN model and the data processing module operate in close integration: for each new session, a feature vector is generated, the model.predict function is invoked, and the classification result is obtained.

Monitoring and Alerting Module — a component intended to present the system's results to information security operators and/or other external systems. In the implemented version, it consists of two parts:

–*Logical alerting module*, which makes decisions based on the outputs of Suricata and the ANN. Several implementation options are possible: the simplest approach is to output all detections (both signature-based and ANN-based) into a single stream; a more advanced option involves cross-correlation of results (for example, generating a high-priority alert when both Suricata and the ANN produce the same conclusion, or forwarding an event for additional verification when the results differ); alert suppression is also possible if the ANN classifies an event as a false positive. In the current implementation, the principle “any indication of an attack is sufficient for alerting” is adopted: if Suricata generates an alert, it is recorded (with the corresponding signature); if Suricata does not detect an attack but the ANN classifies the session as an attack, an anomaly alert is generated (with the attack type identified by the ANN); if both components detect an attack, the alert is strengthened (marked as confirmed by both signature-based and intelligent methods).

–*Visualization and reporting interface*, which provides operators with the ability to observe events. For this purpose, a web interface based on FastAPI (a lightweight Python framework for building APIs) and a simple web page were implemented. When an event is detected, the monitoring module records it in storage (in memory or in a file) and provides access to the current list of incidents via an API. The web page periodically queries the API and updates the incident table. For each suspicious session, the following information is displayed: timestamp, source and destination (IP

address, port), Suricata decision (if a signature alert is present, the corresponding signature), ANN decision (attack class or a «normal traffic» assessment with probabilities), and the final system decision. The interface allows filtering and sorting of data by time and event type. Furthermore, basic charts are used for showing the amount of occurrences across attack classes, thus allowing the user to notice anomalous increase in activity.

The relationship among the system elements is the following: logs of the network session are recorded by Suricata permanently; the Data Processing module watches the log file and when a completed session log appears, extracts features out of it; then the information is sent to the ANN for classification in a few milliseconds, after which the monitoring module generates the event. Therefore, in fact, the processing takes place as near real-time processing but the overall waiting time depends on session buffering time (that is either the session has to end or timeout). In case of long-living sessions it is possible to make an intermediate evaluation by sending the data to the ANN every N seconds.

In the system being used, the focus is specifically on brief TCP and UDP sessions. This is because most attacks only take seconds or minutes to take place. Thus, the worst-case detection times are equal to the session length. This is okay for attacks like scanning or DoS. However for rapid long-duration attacks, like slow infiltration, further optimization is possible through analyzing traffic data with a lower aggregation level and/or configuring Suricata to generate intermediate log records.

From the computational point of view, the architecture is implemented as follows: Suricata operates on a dedicated virtual machine (hereinafter referred to as VM) operating under Ubuntu, with a separate CPU core assigned for packet capturing, thus providing the possibility to monitor the traffic at approximately 1 Gbps. The ANN component and data processing are based on another VM (Monitoring VM). The components communicate with each other over the network. Suricata writes its logs into a shared directory that is accessible from the Monitoring VM, or alternatively can stream through ZeroMQ (Suricata supports streaming of Eve logs). In this case, the technology called shared log file is used. The Monitoring VM has a GPU (NVIDIA 1080 in the experimental environment), which speeds up the ANN inference process. Nevertheless, with smaller models, when the ANN can operate with the CPU, CPU can also be used. Interactive interface (FastAPI and the web page) is hosted at the Monitoring VM.

From a fault tolerance perspective, the system can function in passive mode: if ANN does not work properly, the Suricata continues its operation of logging and generating signature-based alerts without relying on the intelligent component at all. In case Suricata becomes unavailable, ANN will not have fresh information, but will still be operational and will not return any new events. Therefore, components are not dependent on each other strictly, increasing reliability of the entire system. It is worth noting that the communication methods used (files, REST API) are simple and easy to substitute.

The integration of the system into IT infrastructure assumes that Monitoring VM will be placed in Security Administrators segment. Monitoring module is able to push the alerts to the external systems, such as SIEM system via Syslog/CEF or via email. During the project, a simple notification scheme was set up: in case of critical attacks alert messages are logged into a separate file and sent via email (for demo purposes using SMTP). Simple graphs display number of events based on the type of attack to assist operators in recognizing sudden spikes in the activity.

The operation of the system occurs in the following manner: Suricata continues to log the network sessions continuously and the Data Processing module is responsible for monitoring the log file; when a session completes, it processes the logging file in order to pull out useful features. After a few milliseconds, the features reach the ANN for classification. When the input from the ANN is received, the Monitoring module generates the event. Almost all actions take place in real-time, with the only delays resulting from waiting for the session to complete or time out so Suricata indicates the stats. For long sessions occasionally reaching to the ANN every few seconds is possible.

This system aims at short TCP/UDP sessions since in most cases attacks occur within seconds or minutes. Thus, in the worst case scenario, the delay for detection is equal to the session length, making the system suitable for identifying attacks such as scans and DoS attacks. When it comes to

longer and slower attacks, however, it is necessary to use some additional techniques such as conducting traffic analysis on more detailed levels or receiving logs from the Suricata in the process.

Suricata is hosted on an Ubuntu virtual machine (VM) that uses its own CPU core for packet capturing and can handle approximately 1 Gbps of traffic. ANN and data processing are hosted on the Monitoring VM. The two VMs communicate with one another through the network: Suricata sends logs to a shared directory, which Monitoring VM monitors, or alternatively, logs can be streamed to the Monitoring VM via ZeroMQ. In the existing setup, logs are exchanged as files. Monitoring VM has a GPU (NVIDIA 1080 in the test system) to accelerate ANN, although smaller models can work on the CPU only. The user interface is implemented with FastAPI and runs in a browser on the same VM.

If anything malfunctions, the system continues to operate in a rudimentary way. For example, if ANN fails, Suricata still logs the data and generates signature-based alerts. If Suricata stops logging, ANN does not get any new data but remains alive without producing any new events. This loose coupling of parts increases the reliability of the entire system. The means of communication, such as files and REST API, are obvious and can be easily replaced.

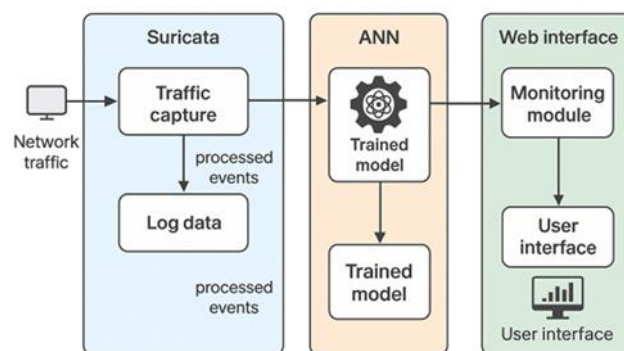


Figure 2. Overall system architecture

In conclusion, it should be noted that the presented overall architecture corresponds to a typical hybrid IDS scheme: Suricata acts as a sensor, generating enriched data; the neural network performs intelligent analysis of these data; and the monitoring module integrates the obtained results and provides interaction with the user. All system components are implemented according to a modular principle, which makes it possible, if necessary, to replace Suricata with Snort (with minor changes in data processing logic) or to update or retrain the ANN model without modifying the other parts of the system. This approach ensures the scalability and flexibility of the architecture.

Results and its discussion

In general, the proposed design is based on the principles of a typical hybrid IDS. Suricata fulfills the function of a sensor that generates enriched data, while the neural network is used to analyze this information, whereas the monitoring module is responsible for merging the results and interacting with users. The components of the system are modular, which allows for the replacement of Suricata with Snort with only slight changes in logic and retraining of the ANN model without altering other parts of the system.

Intrusion detection can be straightforward, where normal traffic is distinguished from intrusion attempts, or it can be complicated. More advanced systems need to distinguish not only whether an intrusion is detected but also its type, such as port scanning, DDoS, web application attack, or insider threat. This is an example of multiclass classification, which refers to classification problems that involve more than two classes.

In our case, based on the *CICIDS2018* dataset, several categories of network attacks were identified, including Brute Force, DoS, DDoS, PortScan, Bot, Web Attack, Infiltration, and others, as well as the «Legitimate» class corresponding to normal traffic. In the general case, for example, $K = 8$ classes are considered.

Multiclass attack classification imposes specific requirements on model design and evaluation.

For (K) classes, an output layer consisting of (K) neurons with a softmax activation function is typically used, enabling the model to produce a probability distribution over the classes. During training, categorical cross-entropy is commonly employed as the loss function.

$$L = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K y_{n,k} \log(\hat{p}_{n,k})$$

where $y_{n,k}$ is the label equal to 1 if the n-th sample belongs to class k, and 0 otherwise;

$\hat{p}_{n,k}$ is the probability predicted by the model, i.e., the probability that the nnn-th sample belongs to class k.

Cross-entropy measures the divergence between the true distribution (concentrated on the actual class) and the probability distribution predicted by the model. Minimizing this function encourages the probability assigned to the correct class to approach 1, while the probabilities of the remaining classes approach 0.

In multiclass problems, it is common for the number of samples in some classes to significantly exceed that in others. In our dataset, for example, the number of normal sessions substantially exceeds the number of sessions corresponding to rare attack types, which reflects real networks where most traffic is legitimate. Thus, classes such as DoS and PortScan may have a large number of examples, whereas the Infiltration class, corresponding to complex attacks, is represented by a very limited number of samples. This can lead to a bias of the model toward more frequent classes, since the contribution of errors on rare classes to the overall loss function is relatively small. To address this issue, as noted earlier, class weighting and batch balancing were applied during training. By increasing the penalty for misclassification of rare attacks, the model was encouraged to pay greater attention to their characteristic features. As a result, acceptable values of quality metrics were achieved even for classes with an extremely small number of examples.

When evaluating model performance in multiclass classification settings, the simple metric of accuracy—the proportion of correctly classified samples—does not always reflect real effectiveness, especially in the presence of class imbalance. For example, if 95% of the traffic is normal and the model always predicts the «normal» class, it will achieve an accuracy of about 95% while being practically useless, as it fails to detect attacks. Therefore, metrics that account for classification quality for each class separately are used, such as precision, recall, and F1-score, calculated for each class, as well as their averaged values (macro- and weighted metrics).

Precision for class A is defined as

$$A = \frac{TP_A}{TP_A + FP_A}$$

where TP_A is the number of sessions belonging to class A that are correctly classified by the model as A;

FP_A is the number of sessions belonging to other classes that are incorrectly classified by the model as class A. In the context of an intrusion detection system, if class AAA corresponds to the “attack” class, the precision metric indicates the proportion of events classified as attacks that are indeed attacks, and thus characterizes the level of false positives.

Recall for class A is defined as

$$A = \frac{TP_A}{TP_A + FN_A}$$

where FN_A is the number of sessions belonging to class A that were incorrectly classified by the model as another class (missed cases). Recall indicates the proportion of all real attacks of class A that the model was able to detect and effectively reflects the sensitivity or detection

capability of the system.

The *F1-score* is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

The F1-score serves as a generalized indicator of model performance and is particularly useful in the presence of class imbalance, as it simultaneously accounts for both the number of false positives and the number of missed attacks.

In multiclass classification, precision, recall, and F1-score are computed for each class (with the class under consideration treated as the «positive» class and all other classes treated as “negative” within the scope of that evaluation). Then, as a rule, averaged metrics such as macro-average and weighted-average are reported. The macro-average represents the mean value of a metric across all classes, obtained by simple averaging (i.e., all classes are considered equally important). The weighted-average is calculated as a weighted mean of the metrics across classes, with weights proportional to the number of samples in each class (i.e., more frequent classes contribute more).

The report presents results for all the above metrics. Particular attention is paid to the aggregated «Attack» class (i.e., combining all attack types into a single class versus normal traffic), as this allows assessment of the overall ability of the system to detect intrusions. For such a binary formulation, ROC curves and the AUC metric can be used; however, in multiclass classification settings, ROC/AUC is not always the most informative, since these metrics are primarily applicable to binary problems.

Confusion Matrix. For a clear analysis of results, a confusion matrix of size $K \times K$ is often used, where rows correspond to true classes and columns correspond to predicted classes. The element at position (i, j) indicates how many sessions belonging to class i were classified by the model as class j. In the ideal case, the matrix is diagonal (all instances are classified correctly). Analysis of off-diagonal elements makes it possible to identify the most frequent types of errors. In the context of an IDS, the confusion matrix may, for example, show that a significant proportion of misclassifications occur between attack types with similar features: thus, DoS and DDoS are often confused with each other (the model detects a load-related anomaly but does not always distinguish between a single-source DoS and a distributed DDoS). In addition, web application attacks may be incorrectly classified as normal traffic (false negatives), which may indicate weakly expressed features of web attacks and the need to improve them. In the present work, such a matrix was constructed and presented for a detailed analysis of the error characteristics of multiclass classification.

Training on imbalanced data. If w_k denotes the weight corresponding to class k , then the loss function is modified as follows:

$$L = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K w_k y_{n,k} \log(\hat{p}_{n,k})$$

The larger the value of w_k , the more strongly the network is penalized for an error associated with class k , which corresponds to increasing the recall value for this class. In this work, class weights were set inversely proportional to their frequency in the training data, for example:

$$w_k = \frac{1}{freq_k}$$

As a result, the model paid greater attention to rarely occurring attack categories, which made it possible to improve the quality of their detection.

Interpretation of multiclass results. Unlike the binary case, a single precision or recall value cannot adequately characterize the performance of the entire system; in multiclass classification, it is necessary to analyze a set of metrics. The priority of classes may vary depending on the context: for

example, detecting rare but dangerous attacks (such as infiltration) may be more critical than identifying large-scale scanning activity. In some studies, additional class importance coefficients are introduced; however, in this research all attack types are considered undesirable, and therefore the main objective is to achieve high F1-score values for all classes.

Example of results. Suppose that the model achieved the following indicators: Precision_{macro} = 0.96, Recall_{macro} = 0.95, F1_{macro} = 0.95. At the same time, the following values were obtained for individual classes: normal traffic (P = 0.98, R = 0.99), DoS (P = 0.97, R = 0.96), PortScan (P = 0.99, R = 0.99), Web attack (P = 0.90, R = 0.85), Infiltration (P = 0.85, R = 0.80), and others. As can be seen, for some rare and more complex attacks the metrics are lower, which is expected due to the limited amount of data and the high complexity of the corresponding scenarios. Nevertheless, with F1 values above 0.8 for all classes, the system as a whole can be considered effective. Achieving such results was precisely the goal of this study.

Multiclass classification allows the system not only to detect the presence of an intrusion but also to identify the specific type of attack, which is of significant practical value, since different attacks require different response measures. At the same time, the multiclass approach is more complex: it is necessary to ensure high detection recall for all significant classes while simultaneously controlling the level of false positives for each attack type. The proposed ANN model, trained and tuned using the described methods, demonstrated high overall quality indicators at the testing stage (accuracy and macro-F1 on the order of 0.95–0.97), as well as acceptable metric values for complex and rare classes (in no class did the F1-score fall below approximately 0.8).

Table 1. Comparison of attack detection methods performance

Method	Precision	Recall	F1-score
Suricata (signature-based analysis)	0.92	0.88	0.90
(ANN (neural network model)	0.94	0.93	0.93
Hybrid system (Suricata + ANN)	0.97	0.96	0.96

Conclusion

The scientific novelty of this study lies in the development and experimental validation of a hybrid network intrusion detection architecture that combines signature-based analysis using the Suricata IDS with a multiclass neural network model for intelligent network traffic analysis. Unlike existing approaches, this work implements the integration of signature-based and neural network solutions at the level of result correlation, which makes it possible to increase attack detection coverage and reduce the number of false positives without abandoning traditional security mechanisms.

The authors' contribution includes the design of a modular hybrid IDS architecture, the development of a network data processing pipeline, the training and tuning of a neural network model with consideration of class imbalance, and the experimental evaluation of the effectiveness of the proposed solution under multiclass network traffic classification conditions.

References

1. Diana L., Dini P., Paolini D. Overview on intrusion detection systems for computers networking security. Computers, – 2025. – Vol. 14, no. 3. – Art. 87. – DOI: [10.3390/computers14030087](https://doi.org/10.3390/computers14030087).
2. Buczak A. L., Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, – 2016. – Vol. 18, no. 2. – P. 1153–1176. – DOI: [10.1109/COMST.2015.2494502](https://doi.org/10.1109/COMST.2015.2494502)
3. Dini P., Elhanashi A., Begni A., Saponara S., Zheng Q., Gasmi K. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. Applied Sciences, – 2023. – Vol. 13, no. 13. – Art. 7507. – DOI: [10.3390/app13137507](https://doi.org/10.3390/app13137507).
4. Arp D., Quiring E., Pendlebury F., Warnecke A., Pierazzi F., Wressnegger C., Cavallaro L.,

Rieck, K. Dos and don'ts of machine learning in computer security. Proceedings of the 31st USENIX Security Symposium (USENIX Security 22). – Boston : USENIX Association, 2022. – P. 3971–3988. https://www.usenix.org/system/files/sec22summer_ar_p.pdf

5. Laptiev O., Musienko A., Nakonechnyi V., Sobchuk A., Gakhov S. O. Algorithm for recognition of network traffic anomalies based on artificial intelligence. Proceedings of the 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). – IEEE, 2023. – P. 1–5. – DOI: [10.1109/HORA58378.2023.10156702](https://doi.org/10.1109/HORA58378.2023.10156702).

6. Arp D., Quiring E., Pendlebury F., Warnecke A., Pierazzi F., Wressnegger C., Cavallaro L., Rieck K.. Pitfalls in machine learning for computer security. Communications of the ACM. – 2024. – Vol. 67, no. 11. – P. 104–112. – DOI: [10.1145/3643456](https://doi.org/10.1145/3643456).

7. Verma P., Dumka A., Singh R., Ashok A., Gehlot A., Malik P. K., Gaba G. S., Hedabou M. A novel intrusion detection approach using machine learning ensemble for IoT environments. – 2021. – Vol. 11, no. 21. – Art. 10268. – DOI: [10.3390/app112110268](https://doi.org/10.3390/app112110268).

8. Drăgușin S.-A., Boștinăru R.-N., Bizon N., Iana G.-V. AI-based embedded framework for cyber-attack detection through signal processing and anomaly analysis. Applied Sciences. – 2026. – Vol. 16, no. 3. – Art. 1416. – DOI: [10.3390/app16031416](https://doi.org/10.3390/app16031416).

9. Bhattacharya S., Khanna A., Ganapaneni S., Najana M. Attention-based deep learning frameworks for network intrusion detection: an empirical study. International Journal of Global Innovations and Solutions. – 2024. – DOI: [10.21428/e90189c8.eb32676c](https://doi.org/10.21428/e90189c8.eb32676c).

10. Awajan A. A novel deep learning-based intrusion detection system for IoT networks. Computers, – 2023. – Vol. 12, no. 2. – Art. 34. – DOI: [10.3390/computers12020034](https://doi.org/10.3390/computers12020034).

11. Hossain Md. A. Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach. EURASIP Journal on Information Security. – 2025. — Vol. 2025. — Article 28. – DOI: [10.1186/s13635-025-00202-w](https://doi.org/10.1186/s13635-025-00202-w).

12. Mitra A. Shekhar S. A multiclass approach for network intrusion detection using convolutional neural networks. International Journal of Scientific Development and Research. – 2020. – Vol. 5, no. 4. – P. 253-262. <https://ijsdr.org/papers/IJSDR2004044.pdf>

13. Fatima R., Sheeba S., Rafa R. AI-driven intrusion detection system: a survey of techniques, datasets, and evaluation frameworks. Journal of Systems Engineering and Electronics. – 2025. 35. 212. <https://doi.org/10.5281/zenodo.17949358>.

14. Yin C., Zhu Y., Fei J., He X., A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. – 2017. – Vol. 5. – P. 21954–21961. – DOI: [10.1109/ACCESS.2017.2762418](https://doi.org/10.1109/ACCESS.2017.2762418).

15. Muneer S., Farooq U., Athar A., Raza M. A., Ghazal T. M., Sakib S. A critical review of artificial intelligence-based approaches in intrusion detection: a comprehensive analysis. Journal of Engineering. – 2024. – Vol. 2024. – Art. 3909173. – DOI: [10.1155/2024/3909173](https://doi.org/10.1155/2024/3909173).

16. Ali M. L., Thakur K., Schmeelk S., Debello J., Dragos D. Deep learning vs. machine learning for intrusion detection in computer networks: a comparative study. Applied Sciences, – 2025. – Vol. 15, no. 4. – Art. 1903. – DOI: [10.3390/app15041903](https://doi.org/10.3390/app15041903).

Әдебиеттер тізімі

1. Diana L., Dini P., Paolini D. Overview on intrusion detection systems for computers networking security. Computers, – 2025. – Vol. 14, no. 3. – Art. 87. – DOI: [10.3390/computers14030087](https://doi.org/10.3390/computers14030087).

2. Buczak A. L., Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, – 2016. – Vol. 18, no. 2. – P. 1153–1176. – DOI: [10.1109/COMST.2015.2494502](https://doi.org/10.1109/COMST.2015.2494502)

3. Dini P., Elhanashi A., Begni A., Saponara S., Zheng Q., Gasmi K. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. Applied Sciences, – 2023. – Vol. 13, no. 13. – Art. 7507. – DOI: [10.3390/app13137507](https://doi.org/10.3390/app13137507).

4. Arp D., Quiring E., Pendlebury F., Warnecke A., Pierazzi F., Wressnegger C., Cavallaro L., Rieck, K. Dos and don'ts of machine learning in computer security. Proceedings of the 31st USENIX

Security Symposium (USENIX Security 22). – Boston : USENIX Association, 2022. – P. 3971–3988.
https://www.usenix.org/system/files/sec22summer_arp.pdf

5. Laptiev O., Musienko A., Nakonechnyi V., Sobchuk A., Gakhov S. O. Algorithm for recognition of network traffic anomalies based on artificial intelligence. Proceedings of the 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). – IEEE, 2023. – P. 1–5. – DOI: [10.1109/HORA58378.2023.10156702](https://doi.org/10.1109/HORA58378.2023.10156702).

6. Arp D., Quiring E., Pendlebury F., Warnecke A., Pierazzi F., Wressnegger C., Cavallaro L., Rieck K.. Pitfalls in machine learning for computer security. Communications of the ACM. – 2024. – Vol. 67, no. 11. – P. 104–112. – DOI: [10.1145/3643456](https://doi.org/10.1145/3643456).

7. Verma P., Dumka A., Singh R., Ashok A., Gehlot A., Malik P. K., Gaba G. S., Hedabou M. A novel intrusion detection approach using machine learning ensemble for IoT environments. – 2021. – Vol. 11, no. 21. – Art. 10268. – DOI: [10.3390/app112110268](https://doi.org/10.3390/app112110268).

8. Drăgușin S.-A., Boștinariu R.-N., Bizon N., Iana G.-V. AI-based embedded framework for cyber-attack detection through signal processing and anomaly analysis. Applied Sciences. – 2026. – Vol. 16, no. 3. – Art. 1416. – DOI: [10.3390/app16031416](https://doi.org/10.3390/app16031416).

9. Bhattacharya S., Khanna A., Ganapaneni S., Najana M. Attention-based deep learning frameworks for network intrusion detection: an empirical study. International Journal of Global Innovations and Solutions. – 2024. – DOI: [10.21428/e90189c8.eb32676c](https://doi.org/10.21428/e90189c8.eb32676c).

10. Awajan A. A novel deep learning-based intrusion detection system for IoT networks. Computers, – 2023. – Vol. 12, no. 2. – Art. 34. – DOI: [10.3390/computers12020034](https://doi.org/10.3390/computers12020034).

11. Hossain Md. A. Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach. EURASIP Journal on Information Security. – 2025. — Vol. 2025. — Article 28. – DOI: [10.1186/s13635-025-00202-w](https://doi.org/10.1186/s13635-025-00202-w).

12. Mitra A. Shekhar S. A multiclass approach for network intrusion detection using convolutional neural networks. International Journal of Scientific Development and Research. – 2020. – Vol. 5, no. 4. – P. 253-262. <https://ijsdr.org/papers/IJSDR2004044.pdf>

13. Fatima R., Sheeba S., Rafa R. AI-driven intrusion detection system: a survey of techniques, datasets, and evaluation frameworks. Journal of Systems Engineering and Electronics. – 2025. 35. 212. <https://doi.org/10.5281/zenodo.17949358> .

14. Yin C., Zhu Y., Fei J., He X., A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. – 2017. – Vol. 5. – P. 21954–21961. – DOI: [10.1109/ACCESS.2017.2762418](https://doi.org/10.1109/ACCESS.2017.2762418).

15. Muneer S., Farooq U., Athar A., Raza M. A., Ghazal T. M., Sakib S. A critical review of artificial intelligence-based approaches in intrusion detection: a comprehensive analysis. Journal of Engineering. – 2024. – Vol. 2024. – Art. 3909173. – DOI: [10.1155/2024/3909173](https://doi.org/10.1155/2024/3909173).

16. Ali M. L., Thakur K., Schmeelk S., Debello J., Dragos D. Deep learning vs. machine learning for intrusion detection in computer networks: a comparative study. Applied Sciences, – 2025. – Vol. 15, no. 4. – Art. 1903. – DOI: [10.3390/app15041903](https://doi.org/10.3390/app15041903).

СИГНАТУРАЛЫҚ ТАЛДАУ ЖӘНЕ МАШИНАЛЫҚ ОҚЫТУ ӘДІСТЕРІ НЕГІЗІНДЕГІ ЖЕЛІЛІК ШАБУЫЛДАРДЫ АНЫҚТАУДЫҢ ГИБРИДТІ ЖҮЙЕСІ

ИГИСИНОВ С.Ж.¹, РЫСКЕЛДІҚЫЗЫ Р.¹, ГАЙСАГАЛЕЕВА Б.М.²,
УВАЛИЕВА Г.М.³, ЖУМАБАЕВА Ж.О.^{4*}

Игисинов Сабит Жандарбекович¹ – PhD, математика кафедрасының қауымдастырылған профессор м.а., М.Х. Дулати атындағы Тараз университеті, Тараз қ., Қазақстан

E-mail: igisnovsabit@gmail.com, <https://orcid.org/0000-0002-4189-9142>

Рыскелдіқызы Рабиға¹ – Магистр, математика кафедрасының аға оқытушысы, М.Х. Дулати атындағы Тараз университеті, Тараз қ., Қазақстан

E-mail: makulbekova.1985@mail.ru, <https://orcid.org/0009-0003-0076-203X>

Гайсагалеева Бахит Мухитовна² – Магистр, информатика пәнінің мұғалімі, Ақтөбе қаласындағы Назарбаев Зияткерлік мектебінің жаратылыстану-математикалық бағытындағы, Ақтөбе қ., Қазақстан

E-mail: gaisagaleeva_m@akb.nis.edu.kz, <https://orcid.org/0009-0001-2419-0414>

Увалиева Гүлсия Максатовна³ – 2 курс магистранты, компьютерлік технологиялар және жасанды интеллект кафедрасы, Ш. Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университеті, Ақтау к., Қазақстан

E-mail: w_g_w@mail.ru, <https://orcid.org/0009-0008-6151-0457>.

*Жумабаева Жанар Орынбасаровна⁴ – Техника ғылымдарының магистрі, М. Әуезов атындағы Оңтүстік Қазақстан зерттеу университеті, Шымкент к., Қазақстан.

E-mail: janaru_77@bk.ru, <https://orcid.org/0009-0007-8906-1801>

Аңдатпа. Қазіргі заманғы желілік инфрақұрылымдардың күрделілігінің артуы, желілік трафиктің динамикалық сипаты және кибершабуылдар санының өсуі жағдайында желіге ену әрекеттерін анықтау жүйелерінің тиімділігін арттыру өзекті міндет болып табылады. Зерттеудің мақсаты – сигнатуралық талдауды жүзеге асыратын Suricata жүйесі мен желілік трафикті интеллектуалды талдауға арналған жасанды нейрондық желіні біріктіретін гибриді шабуылдарды анықтау жүйесін әзірлеу және эксперименттік бағалау. Зерттеу барысында аномалияларды анықтау жүйелеріне қойылатын талаптар талданып, қолданыстағы шешімдер салыстырылды, модульдік архитектура жобаланды және CICIDS2018 деректер жинағы негізінде желілік трафикті көпкласты жіктеу бойынша эксперименттік модельдеу жүргізілді. Ұсынылған жүйе желілік трафикті жинау, деректерді алдын ала өңдеу, нейрондық желі арқылы жіктеу, мониторинг жүргізу және ескерту хабарламаларын қалыптастыру модульдерінен тұрады. Кластар арасындағы деректер теңгерімсіздігінің әсерін азайту үшін кластарды салмақтау және оқыту деректерін теңгеру әдістері қолданылды. Эксперимент нәтижелері сигнатуралық және интеллектуалды талдауды біріктіру шабуылдарды анықтау тиімділігін арттыратынын көрсетті. Suricata + ANN гибриді жүйесі үшін Precision, Recall және F1-score мәндері сәйкесінше 0,97, 0,96 және 0,96 болды. Зерттеудің ғылыми жаңалығы сигнатуралық және нейрондық талдау нәтижелерін біріктіру арқылы шабуылдарды анықтау ауқымын кеңейтуде көрінеді. Ұсынылған архитектураны корпоративтік желілерді нақты уақытқа жақын режимде мониторингтеу және қорғау үшін қолдануға болады.

Түйін сөздер: ақпараттық қауіпсіздік, ену әрекеттерін анықтау, желілік трафик, аномалиялар, машиналық оқыту, нейрондық желілер, шабуылдарды жіктеу.

ГИБРИДНАЯ СИСТЕМА ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК НА ОСНОВЕ СИГНАТУРНОГО АНАЛИЗА И МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

ИГИСИНОВ С.Ж.¹, РЫСКЕЛДІҚЫЗЫ Р.¹, ГАЙСАГАЛЕЕВА Б.М.²,
УВАЛИЕВА Г.М.³, ЖУМАБАЕВА Ж.О.^{4*}

Игисинов Сабит Жандарбекович¹ – PhD, и.о. ассоциированного профессора кафедры математики, Таразский университет имени М.Х. Дулати, г. Тараз, Казахстан

E-mail: igisinovsabit@gmail.com, <https://orcid.org/0000-0002-4189-9142>

Рыскелдіқызы Рабиға¹ – Магистр, старший преподаватель кафедры математики, Таразский университет имени М.Х. Дулати, г. Тараз, Казахстан

E-mail: makulbekova.1985@mail.ru, <https://orcid.org/0009-0003-0076-203X>

Гайсагалеева Бахит Мухитовна² – Магистр, учитель информатики, Назарбаев Интеллектуальная школа естественно-математического направления города Ақтөбе, г. Ақтөбе, Казахстан

E-mail: gaisagaleeva_m@akb.nis.edu.kz, <https://orcid.org/0009-0001-2419-0414>

Увалиева Гүлсия Максатовна³ – Магистрант 2-курса, кафедра компьютерных технологий и искусственного интеллекта, Каспийский государственный университет технологий и инжиниринга имени Ш. Есенова, г. Ақтау, Казахстан

E-mail: w_g_w@mail.ru, <https://orcid.org/0009-0008-6151-0457>.

*Жумабаева Жанар Орынбасаровна⁴ – Магистр технических наук, Южно-Казахстанский исследовательский университет имени М. Ауэзова, г. Шымкент, Казахстан.

E-mail: janaru_77@bk.ru, <https://orcid.org/0009-0007-8906-1801>

Аннотация. В условиях роста сложности современных сетевых инфраструктур, динамического характера сетевого трафика и увеличения числа кибератак повышение эффективности систем обнаружения вторжений является актуальной задачей. Целью исследования является разработка и экспериментальная оценка гибридной системы обнаружения сетевых атак, объединяющей сигнатурный анализ Suricata и искусственную нейронную сеть для интеллектуального анализа сетевого трафика. В работе проведены анализ требований к системам обнаружения аномалий, сравнительный анализ существующих решений, проектирование модульной архитектуры и экспериментальное моделирование с применением многоклассовой классификации данных набора CICIDS2018. Разработанная система включает модули захвата сетевого трафика, предварительной обработки данных, нейросетевой классификации, мониторинга и формирования предупреждений. Для

повышения качества классификации в условиях дисбаланса классов применены взвешивание классов и балансировка обучающих данных. Экспериментальные результаты показали, что объединение сигнатурного и интеллектуального анализа повышает эффективность обнаружения атак. Для гибридной системы Suricata + ANN значения Precision, Recall и F1-score составили соответственно 0,97, 0,96 и 0,96, что превышает показатели отдельных методов. Научная новизна заключается в интеграции результатов сигнатурного и нейросетевого анализа. Практическая значимость состоит в возможности применения предложенной архитектуры для мониторинга и защиты корпоративных сетей в режиме, близком к реальному времени.

Ключевые слова: информационная безопасность, обнаружение вторжений, сетевой трафик, аномалии, машинное обучение, нейронные сети, классификация атак.