КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.33.1155
UDC 004.85:004.056

Olga Vasylenko
Cand. Sc. (Tech.), Assoc. Prof., Department of Information Security and Nanoelectronics
Zaporizhzhia Polytechnic National University, Zaporizhzhia, Ukraine
ORCID: 0000-0001-6535-3462
drvasylenkoolga@gmail.com

Andrii Korotun[1,2]
Cand. Sc. (Phys. & Math.), Assoc. Prof.,
[1]Head of the Department of Information Security and Nanoelectronics
Zaporizhzhia Polytechnic National University, Zaporizhzhia, Ukraine
[2]Senior Research, Department of Metallic State Theory,
G.V. Kurdyumov Institute for Metal Physics of the NAS of Ukraine, Kyiv, Ukraine
ORCID: 0000-0003-4165-2788
andko@zp.edu.ua

DDoS ATTACK DETECTION SYSTEM

Abstract. The subject matter of this research is the comprehensive analysis and development of an automated network traffic classification system for Distributed Denial of Service (DDoS) attack detection. The study focuses on the transition from traditional signature-based protection paradigms, which are increasingly ineffective against modern, polymorphic, and high-intensity cyber threats – to anomaly-based detection systems powered by artificial intelligence (AI). The goal of the work is to identify effective artificial intelligence algorithms for network traffic analysis and to develop an applied software solution capable of detecting and automatically responding to DDoS attacks to ensure the security of modern information systems. The following tasks were solved in the article: an analysis of modern cyber threats and the limitations of traditional IDS/IPS systems was conducted, highlighting the necessity for adaptive AI-based solutions; a comparative study was performed based on quality metrics for machine learning and deep learning algorithms, specifically: Decision Trees, Random Forest, Support Vector Machines (SVM), and Multilayer Perceptrons (MLP); a program for DDoS attack detection was developed using Python libraries; practical recommendations were provided for the implementation, maintenance, and further improvement of the system within a real-world network infrastructure. The following methods used are: based on intelligent traffic analysis, including data preprocessing, feature engineering, and supervised learning. The methodology involves utilizing the modern CICIDS2017 dataset to establish behavioral baselines and evaluate model performance using key metrics such as Accuracy, Precision, Recall, and the F1-score. The following results were obtained: It was proven that despite the high accuracy of deep learning algorithms, particularly the MLP, their computational complexity and training time make them less suitable for responding to rapid and intense attacks. Instead, the Random Forest algorithm was identified as the optimal solution. The software developed based on this algorithm performs real-time binary traffic classification, visualizes and analyzes the obtained data, and allows for the integration of detection results into dynamic firewall rules. Conclusions: The results indicate that ensemble methods are promising for cybersecurity applications where high accuracy and response speed are critical; specifically, the Random Forest algorithm provides an ideal balance of speed and precision for DDoS detection. The integration of these results in the form of a methodology into the "F5 Cybersecurity and Information Protection" educational program at the Department of Information Security and Nanoelectronics of the National University "Zaporizhzhia Polytechnic" confirms the practical and academic relevance of the research.

Keywords: Artificial intelligence; Internet traffic analysis; Machine and Deep Learning algorithms; IDS/IPS; DDoS; Random Forest; Key metrics for evaluating algorithms.

INTRODUCTION

In the context of the rapid development of digital technologies and global computerization, the volume of data transmitted over networks is growing significantly, but the main problem is not only the volume of data, but

also the speed and diversity of modern network traffic. At the same time, the number of cyber threats is increasing, including DoS and DDoS attacks, data leaks, malware, phishing campaigns and other types of malicious activity. The increase in the complexity of cyber-attacks has made traditional signature-based intrusion detection systems (IDS) largely obsolete against polymorphic threats.

Statement of the problem. There is a need to create effective automated attack detection systems that can adapt to new types of threats in real time, since traditional protection tools such as firewalls, antiviruses and classic intrusion detection systems cannot always effectively recognize new, complex or targeted attacks. This requires the introduction of new approaches to information analysis and protection [1], in particular artificial intelligence (AI). Due to its ability to learn, adapt, detect hidden dependencies in data and make decisions based on the knowledge gained, AI allows for more effective and flexible methods of protecting information systems. The main areas of application of AI in cybersecurity are anomaly detection, adaptive learning, combating complex threats and automating protection. The use of AI for analyzing Internet traffic is of particular value and it also able to autonomously block malicious actions such as DDoS attacks or the spread of malware, as well as isolate compromised devices from the network. Thus, the task arises of researching and selecting the optimal AI algorithm for detecting DDoS attacks.

Analysis of recent research and publications. Current research highlights the superiority of Anomaly-Based Detection over traditional methods. By leveraging ML, systems can establish a baseline of "normalcy" through feature engineering. In [2] authors demonstrate that Deep Learning models, specifically Multi-Layer Perceptrons (MLP), can identify DDoS patterns with higher precision than classic firewalls because they analyze the statistical distribution of packets rather than looking for a specific "malicious string".

AI effectively analyzes large volumes of data (logs, network traffic, user behavioral patterns) to identify potential threats and detect anomalies that may signal attacks. AI creates a profile of "normal" network or user behavior, with significant deviations from this model. For example, if there is a sudden increase in activity at an atypical time, or an attempt to log in from another geographical region (suspicious patterns), the system raises an alarm or blocks this action [3].

AI models update their parameters based on new data (adaptive learning), which allows them to quickly respond to new attack vectors (for example, new viruses or phishing methods). Machine learning algorithms analyze large arrays of malicious code samples, comparing new files with known patterns, and if a new file is 95% similar to already known malware, the AI automatically blocks it [4]. Neural networks are used to analyze complex visual and textual features, for example, to recognize phishing websites disguised as official banking resources, using Deep Learning (DL) methods to combat complex threats. AI study historical data on attacks (their frequency, time, types of targets) and predict periods of increased activity of cybercriminals, i.e. it is capable of predicting attacks. AI implementation examples: automatic Gmail spam filtering (up to 99.9% filtered) [5]; CrowdStrike Falcon application uses neural networks to identify even unknown malicious code [6]; Darktrace is an example of a system that creates "digital immunity" by constantly analyzing network activity [7].

Although deep learning is powerful, classical algorithms remain vital for real-time applications due to their lower computational cost and [8] provide a comparative analysis showing that decision trees (C4.5) and support vector methods (SVM) are particularly effective for binary classification (attack against the usual). Their research shows that while SVMs offer high accuracy in high-dimensional spaces, decision trees are often superior in production environments due to their “understandability”, which allows network administrators to understand the specific logic of blocking a connection.

DDoS attacks are inherently temporal events; they occur over a certain time sequence. The authors of the study [9] argue that traditional machine learning ignores the chronological relationship between packets and propose to focus on long short-term memory (LSTM) networks, a subset of deep learning that can “remember” the state of network traffic for several minutes. This allows for the detection of “low- and slow-level” DDoS attacks, but fast and intense DDoS attacks remain out of the researchers” attention.

A critical bottleneck in AI implementation is the quality of training data. Many systems still rely on the outdated KDD”99 dataset, which lacks modern attack vectors, so it is necessary to choose a universal and modern dataset.

Thus, a number of tasks are actualized:

- to select a universal and relevant data set for training, validating, and testing the model;
- to compare some popular machine learning/deep learning (ML/DL) algorithms for detecting DDoS attacks by metrics and select the optimal one;
- to provide recommendations for implementing a protection system.

The purpose of this work is to study approaches to network traffic analysis using artificial intelligence methods and develop an applied solution for classifying traffic types to detect potentially dangerous activity. The work considers approaches to processing and analyzing network traffic using machine and deep learning

algorithms. Special attention is paid to neural networks, decision trees, support vector methods (SVM), as well as models used in modern intrusion detection systems (IDS/IPS) [10]. The result of the work should be the implementation of a program for detecting DDoS attacks using Python libraries.

RESEARCH METHODOLOGY

Intelligent traffic analysis methodology. Intelligent analysis of Internet traffic helps detect suspicious activities on the network using the algorithm:

1) collection of data about what is happening on the network;
2) pre-processing (filtering) of data;
3) formation of features by determining the most important aspects of traffic (Feature Engineering);
4) training the model;
5) verification and validation of the model;
6) implementation.

For Data Collection passive and active approaches are used. In a passive approach, data is observed and intercepted without intervention. Tools like Wireshark or TCPdump [11] collect "raw" packets in real time, while NetFlow/IPFIX [12] aggregate metadata about network flows (IP addresses, ports, data volume, etc.). Active data collection involves some impact, for example, polling devices via SNMP to obtain data about their download or errors, or using APIs to collect logs from cloud services (AWS CloudTrail, Azure Monitor, etc. [13]).

At the Data Preprocessing stage, the data is converted to a "clean" and easy-to-analyze form in several stages:

1. Cleaning. Duplicates are removed, missing values are filled in, and data that is not valuable for security is filtered out (for example, traffic between internal servers).
2. Normalization. Numerical data is brought to a common scale using Min-Max, Z-Score, etc. methods [14], then encoded (for example, TCP – 1, UDP – 2).
3. Structuring. Organizing data by breaking traffic into time segments (e.g., 5-minute segments), creating tables, time series, etc.

Feature engineering is the process of identifying data characteristics/features that best describe potential threats. For example, the main characteristics for packets are size, transmission rate, protocol (TCP/UDP/ICMP), number of SYN packets (as a sign of port scanning); for flows, this is session duration, volume of incoming/outgoing traffic, geolocation, IP address, etc.; for behavior, this could be user activity at a certain time of day, the number of login attempts per minute. To facilitate analysis, automatic feature extraction methods are sometimes used, for example, PCA for dimensionality reduction, LSTM [15] for time series. Model Training involves loading the chosen algorithm with a data set and iteratively tuning its internal parameters.

Algorithms used for analysis. Decision Tree and Random Forest algorithms were used to classify data [16], SVM and neural network MLP [17] are used to classify attack types (DDoS, phishing). For clustering, K-Means and DBSCAN algorithms were used [18], which allow finding anomalies (data groups that differ from normal behavior. For deep learning, algorithms such as CNN (traffic graph analysis) and RNN/LSTM (attack prediction) were used, which work well with sequential or complex patterns [19]. The DDoS attack detection system is implemented using the Python programming language, which provides high flexibility and a wide set of libraries for data processing, building machine learning models, and visualization.

Selection of model quality metrics. Before implementation, at the Model Evaluation stage, the model is evaluated using such metrics as Accuracy/ Correctness, Precision, Recall and F1-score [20]. Accuracy is suitable for balanced data, but not for cases where there are few attacks (for example, 95% legitimate traffic, 5% attacks). Precision determines what proportion of detected threats turned out to be real. Recall determines how many real threats were found. The generalized F1-score (the "golden mean" between Precision and Recall) allows us to assess how well the model distinguishes between normal and malicious traffic.

For the completeness of the study, it was decided to use these metrics with the model training time together, which creates conditions for the optimal choice of algorithm, since the metrics are in contradiction with the training time. Estimates including Precision, F-metric, and Accuracy are preferably used with the determination of chance levels. The relationships between the concepts of awareness, salience, correlation, and significance, as well as their relationships to completeness and accuracy, are discussed in the article [21]. Work in these areas is included in the plan for further research. In addition, to supplement and deepen the assessment of the quality of the model, it is planned to add the ROC-AUC algorithm, which evaluates a model`s ability to distinguish between classes across all possible thresholds by measuring the Area Under the Receiver Operating Characteristic (ROC) Curve, which plots true positive rate vs false positive rate [22].

CICIDS2017 dataset. To test the models, the study uses the open data set CICIDS2017 [23], which collects network traffic consisting of two parts: normal traffic and attack traffic. While normal traffic is, for example, website browsing, email usage, etc., attack traffic includes various types of cyber threats, such as:

- DDoS – attacks that overload the system with requests;
- PortScan – scanning open ports;
- Brute Force – attempts to guess the password;
- Infiltration – penetration into the network;
- Botnet – management of infected computers.

The CICIDS2017 dataset specifies over 80 different features for each connection, including parameters such as the number of bytes transmitted or received, connection duration, number of incoming and outgoing packets, etc. Thus, CICIDS2017 was chosen because it allows training models not only on “normal” traffic, but also on various attacks, which helps build powerful cyber defense systems.

For the study, the data was splitting as follows:

- training set (70%) – for training the model;
- validation set (15%) – for checking during training;
- test set (15%) – for final evaluation to make sure the model works on new data.

Python tools and libraries for DDoS detection system. The DDoS attack detection system is implemented using the Python programming language, which provides high flexibility and a wide set of libraries for data processing, building machine learning models and visualization, in particular, the following main software libraries were used in the study: Pandas [24] for convenient work with tabular data, in particular for loading, filtering, transforming and merging data sets; Scikit-learn (a set of tools for building and evaluating machine learning models, including classification, cross-validation, feature scaling, etc.); Matplotlib and Seaborn libraries for creating high-quality graphics and visualizations, including charts, error matrices, histograms, heat maps, etc.). For complex classification models, high-level APIs within TensorFlow, particularly Keras, offer a streamlined approach to building, training, and evaluating neural networks [25].

The system architecture is implemented in the DDoS Detection System class, which organizes the processes of data processing, model training, results visualization, and new traffic analysis by connecting a number of Python methods and libraries, each of which performs a specialized function within the entire system:

__init__ – performs initialization of ML and DL models used for traffic classification (Decision Tree, Random Forest, SVM, and Neural Network MPL), built using Keras, which is capable of taking into account complex relationships between features.

load_data – is responsible for loading the CICIDS2017 dataset, which contains real network connections of both normal and malicious types (including DDoS) and performs pre-processing, including removing irrelevant or null columns; converting class labels to binary format ("DDoS" or "Normal"); scaling numerical features using MinMaxScaler, which allows bringing all values to a single range [0,1], which is necessary for the correct operation of many models.

train_models – performs training of each of the four models on the training subset of data. After training, the quality of the models is evaluated using the metrics Accuracy (overall accuracy), Precision (accuracy of positive classifications), Recall (completeness of attack detection), F1-score (harmonic average between precision and recall), and Training time (training time for each model). The results are stored for further analysis and visualization.

display_results – generates a summary table with all metrics for each model. In addition to the table, visual charts (e.g., bar graphs) are created that allow us to quickly assess the performance of each algorithm.

visualize_confusion_matrix – for each model, an error matrix is created that shows the number of correctly and incorrectly classified samples in the "DDoS" and "Normal" categories, which allows us to see the typical model errors and the ratio between False Positives and False Negatives.

detailed_report – this method generates an extended text classification report that contains detailed information about precision, recall, F1-score, etc.

feature_importance – for tree-based models (Decision Tree and Random Forest), the method builds a feature importance graph. This allows us to analyze which network traffic characteristics have the greatest impact on classification, and, accordingly, in detecting attacks.

analyze_new_traffic – the method allows us to load a new traffic sample (for example, from a CSV file) and classify it using a model that is determined to be optimal. Thus, the system can be used not only for offline analysis, but also for early detection of potential attacks

RESEARCH RESULTS

To select the optimal ML/DL algorithm, the performance of four popular algorithms (Decision Tree, Random Forest, Support Vector Machine (SVM) and Multilayer Neural Network (MLP)) was compared using the above criteria/metrics. Decision Tree was chosen because it is a simple and understandable algorithm: it builds a tree where each node checks a condition to "split" the data into subgroups. Its obvious advantage is high speed. Among its drawbacks is a tendency to overfitting, especially on "noisy" data, so we previously limited the tree depth to 10 to avoid overfitting and used entropy as a separation criterion. Random Forest, as an "ensemble" of many decision trees that make decisions together, has higher accuracy than Decision Tree, but accordingly requires more resources (memory, time). Its advantage is also less prone to overfitting/overtraining. We have introduced the presets: the model consists of 100 trees, each tree has a maximum of 15 levels.

SVM algorithm searches for the best "hyperplane" to separate classes, it works well with high-dimensional data, is less prone to overtraining, but becomes too slow if there is a lot of data [26]. The following presets have been introduced: an RBF kernel (radial basis function) was used, the regularization parameter C = 1.0. MLP was chosen because it performs well in finding complex patterns, despite disadvantages such as complexity of setup and need for significant resources [27]. Pre-settings: 2 hidden layers (100 and 50 neurons), ReLU activation, optimizer – Adam.

A fragment of the description of the models used is given below (part of the Python program):

```
class DDoSDetectionSystem:
def __init__(self):
self.models = {
"Decision Tree": DecisionTreeClassifier(random_state=42),
"Random Forest": RandomForestClassifier(random_state=42,
n_estimators=100),
"SVM": SVC(kernel="rbf", probability=True, random_state=42),
"Neural Network": MLPClassifier(hidden_layer_sizes=(100, 50),
max_iter=300,
activation="relu", solver="adam", random_state=42)
}
```

The screenshot of the program operation is shown in Fig. 1.

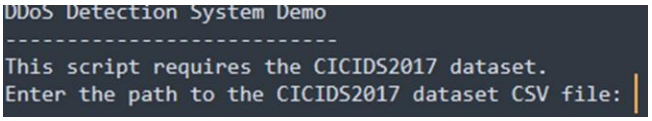

Fig. 1. The program operation

For comparison, the results of the system with the selected algorithms are collected in Table 1.

Table 1

Test results

Metrics	Models			
	Decision Tree	***Random Forest***	***SVM***	***Neural Network***
Accuracy	0.931	0.964	0.902	0.973
Precision	0.928	0.961	0.890	0.975
Recall	0.936	0.968	0.910	0.971
F1-score	0.932	0.964	0.900	0.973
Training time (s)	2.1	3.4	11.6	15.2

As we can see, the neural network showed the highest accuracy and the best F1-score, which means that it detects both attacks and normal traffic well, with almost no errors. The disadvantage is the highest training time compared to other algorithms, which is understandable since this is a complex model with many parameters. SVM turned out to be the least effective, especially when the data has many features (80+). SVM works well for small or medium amounts of data, but as we can see, for large sets it can be slow and not very accurate. Decision Tree is the simplest and fastest algorithm among all tested, but its accuracy was lower, so it is suitable only for non-critical tasks, or when a very fast response is required. Random Forest showed balanced performance, i.e. good accuracy with a fairly high F1-score. This model is trained faster than a neural network and is easier to configure, which makes it a good choice for most practical applications.

Thus, the Random Forest algorithm turned out to be optimal for designing a DDoS attack detection system. After implementing the model into a real network, the model creates rules for IDS/IPS [28], which automatically respond, for example, block IP addresses during an attack. To speed up the work, optimized framework such as TensorFlow Lite (https://www.tensorflow.org) is used, since they work well in real time. There is constant monitoring of errors made and the emergence of new threats, after which the model is retrained.

CONCLUSIONS AND PROSPECTS FOR FURTHER RESEARCH

The result of the research is the practical implementation of a DDoS attack detection system using the Random Forest algorithm trained on the CICIDS2017 dataset. Experimental results confirmed the high efficiency of the chosen approach: the classification accuracy reached 96%, while the training time was one of the lowest. This indicates the promising use of ensemble methods in cybersecurity tasks. It is worth noting that simpler algorithms, despite somewhat lower accuracy, can be effectively applied in real-time systems or on devices with limited computing resources.

Key functionalities of the designed system:

- processing the real dataset;
- binary classification (attack or normal traffic);
- comparison of four models by key metrics;
- visualization of classification results and features;
- generation of error matrices and classification reports;
- ability to analyze new data in real time.

In our opinion, the results of the study may have practical potential for cybersecurity specialists, network administrators, and traffic monitoring system developers. Research aimed at improving intelligent cybersecurity systems is ongoing, and the results are already being used in the educational program "F5 Cybersecurity and Information Protection" at the Department of "Information Security and Nanoelectronics" of the National University "Zaporizhzhia Polytechnic".

The perspective of this research is to develop an active protection system within the framework of the concept of "digital immunity" [29], making the transition from passive detection to active response. By integrating artificial intelligence with software-defined networks (SDN), such systems can not only detect DDoS attacks in real time, but also dynamically reconfigure the network topology to "exit" malicious traffic without human intervention. For proactive defense tasks, it is planned to train the model on CICDDoS2019 [30], which is a specialized, state-of-the-art dataset for various DDoS attack vectors, focusing on modern DDoS methods, such as LDAP, NetBIOS, and MSSQL reflection attacks.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Mpekoa, N. (2024). An analysis of cybersecurity architectures. *International Conference on Cyber Warfare and Security*, 19(1), 200-207. https://doi.org/10.34190/iccws.19.1.2115
2. Kanimozhi, V., & Uppala, T. P. (2022). A comprehensive study of various machine learning algorithms for network intrusion detection using the CICIDS2017 dataset. *International Journal of Computer Science and Network Security*, 22(3), 115-122. https://doi.org/10.22937/IJCSNS.2022.22.3.15
3. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cybersecurity. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502
4. Ahmadi, M., Urunuela, G., Giacinto, G., Munoz-Gonzalez, L., & Lupu, E. C. (2020). Malware classification using binary image representations and deep learning. *Journal of Information Security and Applications*, 55, 102628. https://doi.org/10.1016/j.jisa.2020.102628
5. Deng, J. (2023). Email spam filtering methods: Comparison and analysis. *Highlights in Science, Engineering and Technology*, 38, 187-198. https://doi.org/10.54097/hset.v38i.5805
6. Clark, J. (n.d.). *Researchers explore contrastive learning for malware detection*. CrowdStrike. https://www.crowdstrike.com/en-us/blog/contrastive-learning-enhance-malware-threat-detection/
7. Darktrace. (n.d.). *How does Darktrace detect threats? AI threat detection*. https://www.darktrace.com/cyber-ai-glossary/darktrace-threat-detection
8. Zekri, M., El Sabagh, S., & Badawy, A. (2024). Evaluation of support vector machines and decision trees in classifying high-volume network traffic. *Journal of Network and Computer Applications*, 221, 89-104.

9. Hussain, F., Abbas, S. G., Shah, G. A., & Piran, M. J. (2025). Deep learning-based intrusion detection systems: A study on LSTM and GRU architectures for DDoS mitigation. *IEEE Access*, 13, 10234-10251. https://doi.org/10.1109/ACCESS.2020.3027937
10. Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G. (2019). Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems. *IEEE Access*, 7, 46595-46620. https://doi.org/10.1109/access.2019.2909807
11. Chapman, C. (2016). Using Wireshark and TCP dump to visualize traffic in network performance and security (pp. 195-225). Elsevier. https://doi.org/10.1016/b978-0-12-803584-9.00007-x
12. Pras, R., Sadre, A., Sperotto, A., Fioreze, D., Hausheer, D., & Schönwälder, J. (2009). Using NetFlow/IPFIX for network management. *Journal of Network and Systems Management*, 17(4), 482-487. https://doi.org/10.1007/s10922-009-9138-0
13. Miryala, N. K. (2024). Cloud performance: A comparative study of AWS vs. Azure. *International Journal of Computer Engineering and Technology*, 15(2), 208-223.
14. Henderi. (2021). Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (k-NN) algorithm. *International Journal of Informatics and Information Systems*, 4(1), 13-20. https://doi.org/10.47738/ijiis.v4i1.73
15. Meng, F., Fu, Y., Lou, F., & Chen, Z. (2017). An effective network attack detection method based on kernel PCA and LSTM-RNN. In *2017 International Conference on Computing Systems and Electronics and Control (ICCSEC)* (pp. 396-400). IEEE. https://doi.org/10.1109/iccsec.2017.8447022
16. Kinasih, N. S., Handayani, A. N., Ardiansah, J. T., & Damanhuri, N. S. (2024). Comparative analysis of decision tree and random forest classifiers for structured data classification. *Scientific Information Technology Letters*, 5(2), 13-24. https://doi.org/10.31763/sitech.v5i2.1746
17. Osowski, S., Siwek, K., & Markiewicz, T. (2004). MLP and SVM networks: A comparative study. In *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG 2004)* (pp. 153-156).
18. Ferdiansyah, F. R., Nugraha, R. W., Sofian, R., Purwanto, H., Saepudin, D., & Andriansyah, E. (2024). Implementation of K-means and DBSCAN algorithms: A bibliometric review. In *Advances in Engineering Research* (pp. 192-202). Atlantis Press. https://doi.org/10.2991/978-94-6463-618-5_21
19. Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2024). A comprehensive overview and comparative analysis on deep learning models. *Journal of Artificial Intelligence*, 6(1), 301–360. https://doi.org/10.32604/jai.2024.054314
20. Rahman, M. S. (2024). Understanding accuracy metrics in machine learning models [Preprint]. ResearchGate. https://doi.org/10.13140/RG.2.2.16140.83841
21. Ward, D. M. (2015). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation [Preprint]. ResearchGate.
22. Li, J. (2024). Area under the ROC curve has the most consistent evaluation for binary classification. *PLOS ONE*, 19(12), Article e0316019. https://doi.org/10.1371/journal.pone.0316019
23. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems [Preprint]. ResearchGate.
24. Pandas Development Team. (2024). *Pandas documentation (Version 2.3.3)*. https://pandas.pydata.org/docs
25. Chollet, F. (2015). *Keras: Deep learning for humans*. https://keras.io
26. Schölkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11), 2758-2765. https://doi.org/10.1109/78.650102
27. Youn, Y. R., & Hong, J. (2024). Optimization of model based on ReLU activation function in MLP neural network model. *International Journal of Advanced Smart Convergence*, 13(2), 80-87. https://doi.org/10.7236/IJASC.2024.13.2.80
28. Abbas, S. H., Naser, W. A. K., & Kadhim, A. A. (2023). Subject review: Intrusion detection system (IDS) and intrusion prevention system (IPS). *Global Journal of Engineering and Technology Advances*, 14(2), 155–158. https://doi.org/10.30574/gjeta.2023.14.2.0031
29. Al-Mousa, A., & Ahmed, M. (2025). Autonomous response systems: Bridging the gap between detection and mitigation in SDN. *Journal of Cyber Security and Mobility*, 14(1), 45-68. https://doi.org/10.13052/jcsm2245-1439.1413
30. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2024). Developing a realistic dataset for AI-based DDoS detection: Challenges and methodologies. *Computers & Security*, 136, 103542.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023

Василенко Ольга Валентинівна
к.т.н, доцент, доцент кафедри інформаційної безпеки та наноелектроніки
Національний університет «Запорізька політехніка», Запоріжжя, Україна
ORCID:0000-0001-6535-3462
drvasylenkoolgal@gmail.com

Коротун Андрій Віталійович[1,2]
к.ф.-м.н, доцент,
[1]завідувач кафедри інформаційної безпеки та наноелектроніки
Національний університет «Запорізька політехніка», Запоріжжя, Україна
[2]старший науковий співробітник відділу теорії металічного стану
Інститут металофізики ім. Г.В. Курдюмова НАН України
ORCID: 0000-0003-4165-2788
andko@zp.edu.ua

СИСТЕМА ВИЯВЛЕННЯ DDOS-АТАК

Анотація. Предметом цього дослідження є комплексний аналіз та розробка автоматизованої системи класифікації мережевого трафіку, для виявлення DDoS атак. Дослідження зосереджено на переході від традиційних парадигм захисту на основі сигнатур, які стають дедалі менш ефективними проти сучасних, поліморфних та високоінтенсивних кіберзагроз, до систем виявлення на основі аномалій, що працюють на основі штучного інтелекту (ШІ).
Метою роботи є пошук ефективних алгоритмів ШІ до аналізу мережевого трафіку та розробка прикладного програмного рішення, здатного виявляти та автоматично реагувати на DDoS атаки для забезпечення безпеки сучасних інформаційних систем. У статті було вирішено такі завдання: проведено аналіз сучасних кіберзагроз та обмежень традиційних систем IDS/IPS, підкреслюючи необхідність адаптивних рішень на основі ШІ; проведено порівняльне дослідження на основі метрик якості алгоритмів машинного та глибокого навчання: "Дерева рішень", "Випадкового лісу", методу опорних векторів (SVM) та багатошарового перцептрона (MLP). За допомогою бібліотек Python розроблено програму для виявлення DDoS-атак. Надано практичні рекомендації щодо впровадження, підтримки та подальшого вдосконалення системи в реальній мережевій інфраструктурі. Були отримані такі результати: доведено, що незважаючи на високу точність алгоритмів глибокого навчання, зокрема MLP, обчислювальна складність та час навчання роблять їх менш придатними для реагування на швидкі і напружені атаки, натомість алгоритм "Випадковий ліс" був визначений оптимальним рішенням. Розроблене програмне забезпечення на основі цього алгоритму здійснює бінарну класифікацію трафіку в режимі реального часу, візуалізує та аналізує отримані дані та дозволяє інтегрувати результати виявлення в правила динамічного брандмауера. Висновки: Отримані результати свідчать про те, що ансамблеві методи є перспективними для застосувань кібербезпеки, де критично важливими є висока точність та швидкість реагування, зокрема алгоритм "Випадковий ліс" забезпечує баланс швидкості та точності для виявлення DDoS. Інтеграція цих результатів у вигляді методології в освітню програму F5 Кібербезпека та захист інформації по Кафедрі «Інформаційна безпека та наноелектроніка» НУ «Запорізька політехніка» підтверджує практичну та академічну актуальність дослідження.

Ключові слова: Штучний інтелект; аналіз інтернет-трафіку; алгоритми машинного та глибокого навчання; IDS/IPS; DDoS; "Випадковий ліс"; ключові метрики оцінки алгоритмів.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Mpekoa, N. (2024). An analysis of cybersecurity architectures. *International Conference on Cyber Warfare and Security*, 19(1), 200-207. https://doi.org/10.34190/iccws.19.1.2115
2. Kanimozhi, V., & Uppala, T. P. (2022). A comprehensive study of various machine learning algorithms for network intrusion detection using the CICIDS2017 dataset. *International Journal of Computer Science and Network Security*, 22(3), 115-122. https://doi.org/10.22937/IJCSNS.2022.22.3.15

3. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cybersecurity. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502
4. Ahmadi, M., Urunuela, G., Giacinto, G., Munoz-Gonzalez, L., & Lupu, E. C. (2020). Malware classification using binary image representations and deep learning. *Journal of Information Security and Applications*, 55, 102628. https://doi.org/10.1016/j.jisa.2020.102628
5. Deng, J. (2023). Email spam filtering methods: Comparison and analysis. *Highlights in Science, Engineering and Technology*, 38, 187-198. https://doi.org/10.54097/hset.v38i.5805
6. Clark, J. (n.d.). *Researchers explore contrastive learning for malware detection*. CrowdStrike. https://www.crowdstrike.com/en-us/blog/contrastive-learning-enhance-malware-threat-detection/
7. Darktrace. (n.d.). *How does Darktrace detect threats? AI threat detection*. https://www.darktrace.com/cyber-ai-glossary/darktrace-threat-detection
8. Zekri, M., El Sabagh, S., & Badawy, A. (2024). Evaluation of support vector machines and decision trees in classifying high-volume network traffic. *Journal of Network and Computer Applications*, 221, 89-104.
9. Hussain, F., Abbas, S. G., Shah, G. A., & Piran, M. J. (2025). Deep learning-based intrusion detection systems: A study on LSTM and GRU architectures for DDoS mitigation. *IEEE Access*, 13, 10234-10251. https://doi.org/10.1109/ACCESS.2020.3027937
10. Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G. (2019). Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems. *IEEE Access*, 7, 46595-46620. https://doi.org/10.1109/access.2019.2909807
11. Chapman, C. (2016). Using Wireshark and TCP dump to visualize traffic in network performance and security (pp. 195-225). Elsevier. https://doi.org/10.1016/b978-0-12-803584-9.00007-x
12. Pras, R., Sadre, A., Sperotto, A., Fioreze, D., Hausheer, D., & Schönwälder, J. (2009). Using NetFlow/IPFIX for network management. *Journal of Network and Systems Management*, 17(4), 482-487. https://doi.org/10.1007/s10922-009-9138-0
13. Miryala, N. K. (2024). Cloud performance: A comparative study of AWS vs. Azure. *International Journal of Computer Engineering and Technology*, 15(2), 208-223.
14. Henderi. (2021). Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (k-NN) algorithm. *International Journal of Informatics and Information Systems*, 4(1), 13-20. https://doi.org/10.47738/ijiis.v4i1.73
15. Meng, F., Fu, Y., Lou, F., & Chen, Z. (2017). An effective network attack detection method based on kernel PCA and LSTM-RNN. In *2017 International Conference on Computing Systems and Electronics and Control (ICCSEC)* (pp. 396-400). IEEE. https://doi.org/10.1109/iccsec.2017.8447022
16. Kinasih, N. S., Handayani, A. N., Ardiansah, J. T., & Damanhuri, N. S. (2024). Comparative analysis of decision tree and random forest classifiers for structured data classification. *Scientific Information Technology Letters*, 5(2), 13-24. https://doi.org/10.31763/sitech.v5i2.1746
17. Osowski, S., Siwek, K., & Markiewicz, T. (2004). MLP and SVM networks: A comparative study. In *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG 2004)* (pp. 153-156).
18. Ferdiansyah, F. R., Nugraha, R. W., Sofian, R., Purwanto, H., Saepudin, D., & Andriansyah, E. (2024). Implementation of K-means and DBSCAN algorithms: A bibliometric review. In *Advances in Engineering Research* (pp. 192-202). Atlantis Press. https://doi.org/10.2991/978-94-6463-618-5_21
19. Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2024). A comprehensive overview and comparative analysis on deep learning models. *Journal of Artificial Intelligence*, 6(1), 301–360. https://doi.org/10.32604/jai.2024.054314
20. Rahman, M. S. (2024). Understanding accuracy metrics in machine learning models [Preprint]. ResearchGate. https://doi.org/10.13140/RG.2.2.16140.83841
21. Ward, D. M. (2015). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation [Preprint]. ResearchGate.
22. Li, J. (2024). Area under the ROC curve has the most consistent evaluation for binary classification. *PLOS ONE*, 19(12), Article e0316019. https://doi.org/10.1371/journal.pone.0316019
23. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems [Preprint]. ResearchGate.
24. Pandas Development Team. (2024). *Pandas documentation (Version 2.3.3)*. https://pandas.pydata.org/docs
25. Chollet, F. (2015). *Keras: Deep learning for humans*. https://keras.io
26. Schölkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11), 2758-2765. https://doi.org/10.1109/78.650102

27. Youn, Y. R., & Hong, J. (2024). Optimization of model based on ReLU activation function in MLP neural network model. *International Journal of Advanced Smart Convergence*, 13(2), 80-87. https://doi.org/10.7236/IJASC.2024.13.2.80
28. Abbas, S. H., Naser, W. A. K., & Kadhim, A. A. (2023). Subject review: Intrusion detection system (IDS) and intrusion prevention system (IPS). *Global Journal of Engineering and Technology Advances*, 14(2), 155–158. https://doi.org/10.30574/gjeta.2023.14.2.0031
29. Al-Mousa, A., & Ahmed, M. (2025). Autonomous response systems: Bridging the gap between detection and mitigation in SDN. *Journal of Cyber Security and Mobility*, 14(1), 45-68. https://doi.org/10.13052/jcsm2245-1439.1413
30. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2024). Developing a realistic dataset for AI-based DDoS detection: Challenges and methodologies. *Computers & Security*, 136, 103542.

Отримано редакцією журналу / Received: 12.02.26
Прорецензовано / Revised: 25.02.26
Схвалено до друку / Accepted: 25.06.26