BULLETIN of Cherkasy State Technological University

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

Vol. 31 No. 3. 2026

UDC 004.85:004.056
DOI: 10.62660/bcstu/3.2026.158

Experimental evaluation of machine learning models for detecting cyber threats in network traffic

Ihor Poslied*

Postgraduate Student
National Technical University "Dnipro Polytechnic"
49005, 19 Dmytro Yavornytskyi Ave., Dnipro, Ukraine
https://orcid.org/0009-0006-0885-2007

Abstract. The objective of this study was to increase the ability of automatic systems to identify cyber dangers in network traffic through the use and comparison of various models of machine learning in view of the growing complexity of attacks and the increasing volume of network data. The goal of this research is to determine the features and the effectiveness of standard ML models for the detection of cyber dangers in network traffic by experimental evaluation. This was an experimental, quantitative empirical study in the areas of machine learning and cyber security that was computer based. The research was based on a one-cycle computational experiment on the CIC-IDS2017 dataset, where the models were trained, their performance was evaluated and the patterns in the classification errors were analysed. The results showed a clear relationship between algorithmic complexity and quality of cyber threat detection in network traffic. All the models obtained high values for the different metrics although the ensemble methods always performed better than the baseline approaches. Logistic Regression performed the worst with an accuracy of 94.7%, recall of 92.1% and f1-score of 92.8%. On the other hand, XGBoost achieved the best results, with an accuracy of 99.2%, a recall of 98.9%, an F1-score of 98.9% and an area under the ROC curve of 99.5%, while also causing the fewest misclassifications. Examination of the confusion matrices showed a significant decrease in missed attacks, falling from 3,086 to 437 when XGBoost was used, thus reducing the risk of threats going undetected. The highest classification performance was found for denial-of-service, distributed denial-of-service and port-scanning attacks, with recall reaching 99.8%. Brute-force attacks, botnet activity and web attacks were more difficult to detect because they resembled legitimate traffic, although recall remained above 96% for these categories as well. The results can be used by cybersecurity specialists, developers of intrusion detection and prevention systems and security operations centre analysts

Keywords: ensemble algorithms; XGBoost; confusion matrix; feature importance; intrusion detection and prevention system; distributed denial of service; botnet; CIC-IDS2017

INTRODUCTION

Traditional statistical and signature-based methods of network traffic analysis have a limited capacity to detect new or modified attacks in highly dynamic network environments characterised by large volumes of data. Consequently, there is growing scientific and practical interest in machine learning methods, which can identify latent patterns in multidimensional feature spaces and adapt to changes in network traffic characteristics. However, the effectiveness of these methods depends on the choice of algorithm, the structure of the data, and the characteristics of the attack scenarios. This creates a need for their comprehensive comparative evaluation under identical experimental conditions.

Current research into network intrusion detection is increasingly concentrating on the use of machine learning and deep learning techniques in order to

Article's History: Received: 30.03.2026; Revised: 10.08.2026; Accepted: 21.09.2026; Published: 07.10.2026

Suggested Citation:
Poslied, I. (2026). Experimental evaluation of machine learning models for detecting cyber threats in network traffic. *Bulletin of Cherkasy State Technological University*, 31(3), 158-172. doi: 10.62660/bcstu/3.2026.158.

*Corresponding author (ih_poslied@outlook.com)

Copyright © The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/)

enhance the accuracy of network traffic analysis and to guarantee the timely identification of cyber threats. A. Nikitenko & Y. Sobol (2025) suggested an interactive network intrusion detection system in real-time based on deep learning techniques. They provided evidence that neural networks can be used for automatic analysis of large volumes of network data and showed that such networks have higher accuracy in detecting abnormal activity than traditional methods. Conversely, a detailed review of modern intrusion detection systems (IDSs) for analysing network traffic was conducted by V. Storozenko *et al.* (2025). The authors organised the major architectural approaches to the development of IDSs, described the advantages and disadvantages of both signature-based and anomaly-based detection methods, and highlighted the necessity of incorporating machine learning algorithms for improving the adaptability of these systems. Their analysis revealed that current cyber dangers demand models that can detect previously unknown attack scenarios that cannot be effectively identified by signature-based methods alone. M. Rodriguez *et al.* (2022) took a different approach and did a comparative evaluation of machine learning methods for intrusion detection using network flow features. When the researchers looked at how different classification algorithms performed in detecting network attacks, they discovered that ensemble models offer better accuracy and robustness compared to simpler classifiers when dealing with complex data structures. This finding reinforced the need to select an appropriate machine learning model depending on the characteristics of the dataset and the types of attacks in question.

Research in this area has been focused on the comparison of machine learning algorithms for the detection of network intrusions. M.U. Ilyas & S.A. Alharbi (2022) studied the use of classical and ensemble machine learning algorithms on current internet traffic. They found that models capable of learning the complex non-linear relationships between features have more accuracy in detecting attacks compared to linear classifiers. H.I. Coşar *et al.* (2024) evaluated the performance of different machine learning algorithms on the CSE-CIC-IDS2018 dataset, which contains a variety of modern attacks and network activity scenarios. The results they obtained demonstrated that the ensemble algorithms consistently outperform the baseline models in terms of accuracy, recall and F1-score. T.H. Chua & I. Salam (2023) have evaluated different classifiers using an advanced network traffic dataset and, in this way, have contributed to the study of the performance of machine learning algorithms in the area of network security. They showed that model performance depends greatly on the structure of the data and the type of attacks, while ensemble methods are the most robust with respect to changes in feature distributions.

Another area of research involves the development of models that are efficient in terms of their use of resources for the purpose of detecting distributed denial-of-service (DDoS) attacks. N. Ahmed *et al.* (2026) and A. Elhanashi *et al.* (2022) presented complete machine learning pipelines for the detection of DDoS attacks using the CIC-IDS2018 and CIC-DDoS2019 datasets. N. Ahmed *et al.* (2026) carried out a comparative study of a number of classification algorithms and concluded that optimising feature selection and data preprocessing can considerably lower computational costs without greatly affecting the accuracy of attack detection. On the other hand, the results obtained by A. Elhanashi *et al.* (2022) showed that ensemble methods perform well on complex multiclass datasets and proved that they have a better generalisation ability than the individual baseline classifiers. N. Saini *et al.* (2023) introduced a hybrid ensemble machine learning model for the detection of advanced persistent threat (APT) attacks by examining abnormal network behaviour. The authors proved that combining a number of classification algorithms helps in detecting covert malicious activities which have a long-lasting and low-visibility impact on the network infrastructure. Their results confirmed the benefits of using ensemble methods when it comes to detecting complex cyber threats for which traditional techniques are only partly effective. M.S. Alhakeem & K.B. Ajlan (2024) also carried out a comparative evaluation of machine-learning-based intrusion detection systems in the area of cloud security. Their research showed that the effectiveness of intrusion detection systems relies on the ability of the models to adapt to heterogeneous network traffic and to the complex attack situations typical of current distributed computing environments.

Earlier research has mainly been concerned with creating individual intrusion detection models or with assessing their performance by means of various datasets and specific samples designed to detect certain kinds of attacks. Yet there has not been sufficient attention given to a thorough comparison of classical and ensemble machine learning algorithms under the same experimental conditions when using the CIC-IDS2017 dataset. Moreover, prior studies have offered only limited insight into the structure of classification errors, the detection properties of the different attack classes, and the interpretation of the most significant network traffic features. Because of these limitations, it is difficult to judge the advantages and disadvantages of the various models objectively.

This research aimed to conduct an experimental comparative assessment of various machine learning models for the purpose of detecting cyber threats in network traffic by making use of the CIC-IDS2017 dataset and determining the most effective algorithm on the basis of a number of classification performance measures. In order to achieve this aim, the following objectives were carried out: comparing Logistic Regression, Decision Tree, Random Forest, and XGBoost in terms of accuracy, precision, recall, F1-score, ROC-AUC, and

confusion matrices; investigating how effective these models were at detecting different kinds of network attacks; and analysing the impact of individual network traffic characteristics on the classification results by looking at feature importance.

MATERIALS AND METHODS

It was an experimental study that made use of a computer-based method involving a comparative analysis. In terms of methodology, it was a quantitative empirical study in the areas of machine learning and cybersecurity because it was based on a computational experiment that involved training and testing the models on the standardised CIC-IDS2017 dataset. The experimental part of the study was carried out in one computational cycle after the preprocessing of the data was completed. The entire research process – data preparation, model training and evaluation of the results – was conducted from September 2025 to May 2026 as one continuous and self-contained experimental phase.

The CIC-IDS2017 dataset was created by the Canadian Institute for Cybersecurity in order to offer a realistic foundation for the study and assessment of network intrusion detection techniques. The network traffic consisted of realistic simulations designed to mimic the way a typical corporate network infrastructure functions and included both normal user activity and a variety of attacks. The data were obtained using CICFlowMeter, a tool that generated statistical features at the level of network sessions. The dataset comprised legitimate traffic as well as a wide range of cyber threats that represented the main types of attacks aimed at network infrastructure. The main kinds of malicious activity were DoS, DDoS, brute-force, port-scanning, botnet activity, and web attacks. Each type of attack displayed specific network traffic behaviour patterns which appeared in the corresponding statistical flow features. DoS and DDoS attacks involved a large number of network requests, causing an overload of the target resources. Port scanning was characterised by a large number of short-lived connections to different destination ports, indicating that network reconnaissance was taking place. Brute-force attacks involved repeated authentication attempts, and botnet activity revealed the coordinated behaviour of the compromised hosts. Web attacks involved exploiting vulnerabilities in web applications through traffic using Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) and had a more complex network feature structure.

The dataset comprises over 2.8 million network flows, each with more than 80 statistical features that describe the temporal, transport and statistical characteristics of network interactions. The features were the duration of the flow, the number of packets, the amount of transmitted data, the intervals between the packets, and other aggregated metrics obtained from the analysis of network traffic. The records were distributed among the different classes in a very unbalanced way as it is usually the case for real world network environments. The legitimate traffic (Benign) class was by far the most frequent and the different attack classes represented much smaller proportions depending on the type of threat. This imbalance added more challenges to machine learning algorithms, especially as it increased the chance of the model showing a bias to the majority class and therefore an appropriate method to evaluate the performance of the classification was needed.

In order to improve the quality of the input sample and ensure that the machine learning models were properly trained, data preprocessing was carried out. The first step involved data cleaning, which consisted of changing records containing missing values (NaN), infinite values (±Inf), or invalid numerical values to NaN and then removing them using the replace() and dropna() methods provided by the Pandas and NumPy libraries. The dataset contained anomalous observations which were inconsistent with the physical meaning of the feature values. These anomalies included negative values for parameters that by their nature could not be negative, for example, flow duration and the number of packets or bytes, as well as cases where the network flow duration was zero. Duplicate records were eliminated by means of the drop_duplicates() method. As a result of this cleaning procedure, possible sources of distortion in the statistical properties of the features were removed, and the validity of the following model training was guaranteed. In the second stage, the non-informative and identifying features (Flow ID, Source IP, Destination IP, and Timestamp) were deleted since they did not relate to network traffic behaviour and might have caused overfitting of the model. The numerical features were scaled by means of Min-Max normalisation using the MinMaxScaler function from the Scikit-learn library, thereby transforming all the feature values into the interval [0;1]. This was done so that algorithms which are sensitive to data scale, especially Logistic Regression, could work properly. The target variable was encoded by means of Label Encoding, the Benign class being assigned the value 0 and all the attack types being combined into a single Attack class and given the value 1. This thus created a binary classification problem for the purpose of detecting malicious network activity.

The dataset used in the experiment was arranged so that the machine learning models could be properly trained and their later performance could be objectively evaluated. After the structure of both the training and test data had been decided upon, measures were taken to eliminate any possible bias in the evaluation and to enhance the models' ability to generalise. The first of these measures was to divide the cleaned dataset into a training portion and a test portion. In

order to ensure that the experiment was reproducible and to maintain a proper representation of the classes, stratified data splitting was employed. The training set was used for constructing the models, while the test set was kept solely for the independent assessment of their performance.

Because the original dataset contained an imbalance, the balancing procedure was performed on the training data before the models were developed. At that time, legitimate traffic was the main category, whereas certain types of attack – especially Botnet, Brute Force, and Web Attacks – occurred in only a small number of instances. Once the target variable had been encoded and all the different attack types had been combined into a single Attack class, the Synthetic Minority Oversampling Technique (SMOTE) was applied to the training set in order to correct the imbalance between the Benign and Attack classes. This was achieved by creating synthetic samples for the minority Attack class, which thus balanced the proportions of the classes and reduced the bias of the model towards the more common Benign class. As a result, the training set had a more even distribution of classes, allowing the models to be trained more effectively and decreasing the bias of the classifier towards normal traffic. The models then demonstrated a better ability to detect attacks, as shown by the drop in false negatives from 3,086 with Logistic Regression to 437 with XGBoost. Five-fold cross-validation was applied during the training and selection of the model in order to enhance the reliability of the evaluation and to decrease the reliance of the results on a specific random division of the data. The original dataset was split into k subsets, and the models were then trained and validated by rotating the subset used for validation. The independent test set, which had not been used in the model training, was used to obtain the final numerical results. The accuracy, precision, recall, F1-score and ROC-AUC values cited in the study were calculated from the test set following the completion of training and after the optimal model had been chosen on the basis of the crossvalidation outcomes. This method thus gave rise to stable estimates of performance and an objective evaluation of the models' generalisation ability.

In the experimental study, several classical and ensemble machine learning algorithms were used to perform binary classification on the network traffic extracted from the CIC-IDS2017 dataset. The use of different algorithms was aimed at comparing their effectiveness in detecting cyber dangers in a high-dimensional feature space with non-linear relationships and class imbalance. The baseline model was Logistic Regression, used as a linear classifier to establish a baseline level of performance. This model estimated the probability that an observation was from the Attack class using a sigmoid function of a linear combination of the features. The model itself was simple but provided interpretable results and served as a benchmark for the performance of more complex algorithms. The Decision Tree algorithm is also used which recursively divides the feature space based on the criterion of information entropy or Gini index. This enabled the modelling of the non-linear relationships between the characteristics of the network traffic and the attack class. Decision tree models were highly interpretable but had a tendency to overfit if left too deep.

In order to improve robustness and the ability to generalise, the Random Forest ensemble method was used. This is done by constructing a number of individual Decision Trees and then averaging their predictions. Model variance is reduced, and the stability of the results is increased by combining random feature subspaces with bootstrap samples. The gradient-boosting algorithm called XGBoost was also employed; it trained a series of weak classifiers in each new iteration with the aim of correcting the errors made in the previous one. The model was able to efficiently handle large amounts of data and complex non-linear relationships while at the same time achieving high classification accuracy and being robust against noise in the input features. The main training parameters were chosen on the basis of the nature of the task and the size of the dataset. Logistic Regression employed L2 regularisation with a regularisation parameter set at $C = 1.0$ and was optimised iteratively until convergence. For the Decision Tree, the maximum depth of the tree and the minimum number of samples needed to split a node were limited in order to minimise the risk of overfitting. The importance of the features for both the Random Forest and the XGBoost models was established by means of the methods built into each model for evaluating the contribution of each individual feature to the process of classification. In the case of the Random Forest, Mean Decrease in Impurity (MDI) was employed; this measure referred to the total reduction in node impurity achieved as a result of using a given feature. With respect to XGBoost, Gain was used to indicate the total improvement obtained in the objective function for a feature across all the trees in the ensemble. Feature importance was determined for the final XGBoost model, which had been trained on the entire training set after the data had been preprocessed and the classes balanced. In order to allow a valid comparison, the absolute importance values were normalised by dividing the importance value for each feature by the sum of the importance values for all the features and then multiplying the resulting figure by 100%. The total importance of all the features was thus 100%.

The ability of the machine learning models to detect cyber threats was assessed by means of standard classification metrics, in order that the results might be compared with one another and understood within the context of cybersecurity. Accuracy was defined as the fraction of observations in the entire dataset that

were correctly classified and thus showed the model's general accuracy. Precision referred to the proportion of attacks that were correctly identified out of all the cases that had been labelled as attacks and was used as a measure of false positive predictions. Recall showed how well the model was at detecting all actual attacks and was particularly important in the case of threats that had been missed. To evaluate the classification performance in a balanced way, the F1-score combined precision and recall. To assess the overall discriminatory power of the algorithm, the ROC-AUC was used, which assesses the ability of the algorithm to separate the classes independently of the classification threshold. In addition, a confusion matrix was used to record the number of true positives, true negatives, false positives, and false negatives, so that the structure of the classification errors could be analysed.

The work was implemented in Python language, which made it possible to perform the entire process of data processing, training of the model and evaluation of its performance. The main environment we picked is python, because there are many libraries available to use in machine learning and data analysis. Classical machine learning algorithms were implemented using the Scikit-learn library, which offers tools for preprocessing data, Logistic Regression and Decision Tree models, the Random Forest ensemble method, and classification performance evaluation. We used a special software framework, designed for processing large datasets and dealing with complex non-linear relationships, to implement the XGBoost gradient-boosting algorithm. The employment of this algorithm improved the classification accuracy, and allowed to efficiently handle the high-dimensional feature space of the network traffic. The workstation used for the experiment was a multi-core personal computer with sufficient random-access memory and multithreading support. This configuration allowed the model to be trained stably on the entire CIC-IDS2017 dataset. Thanks to hardware acceleration and the use of optimised versions of the algorithm, the time required to train the ensemble models was decreased, and the experiments could be reproduced. However, the study was constrained by its dependence on a single dataset, namely CIC-IDS2017, and by the fact that only a limited number of classical machine learning models were used, with no attention being paid to deep neural network methods or to actual network traffic streams from production environments.

RESULTS

Comparison of machine learning model performance in detecting network-based cyber threats. The experimental evaluation showed that all the algorithms examined achieved a high degree of classification performance; yet there were significant differences among them with regard to their capability of detecting cyber threats and their ability to reduce misclassifications. As a general rule, performance increased gradually when progressing from the simpler models to the more complex ensemble methods, since these could consider the non-linear relationships among the network traffic characteristics (Table 1).

Table 1. Comparative classification performance of the machine learning models

Model	Accuracy, %	Precision, %	Recall, %	F1-score, %	ROC-AUC, %
Logistic Regression	94.7	94.5	92.1	92.8	95.4
Decision Tree	97.2	96.7	96.1	96.4	97.5
Random Forest	98.9	98.8	98.5	98.6	99.2
XGBoost	99.2	99	98.9	98.9	99.5

Source: compiled by the author

On analysis of the results, it was discovered that Logistic Regression obtained the lowest scores for all the performance measures in question. Even though its accuracy was 94.7%, its recall was only 92.1%, which shows that a fairly large number of cases had been incorrectly classified as negative. The fact that these results were obtained shows that the linear model cannot capture the complex nonlinear patterns which exist in the high-dimensional feature space of network traffic. In cybersecurity, it is particularly important to avoid false-negative predictions of this kind since each such case represents an attack which went undetected and therefore presents a risk to the information system.

The Decision Tree algorithm performed significantly better than Logistic Regression in terms of classification performance. The accuracy was up to 97.2% and the precision, recall and F1-score were all higher than 96%. This indicated that the Decision Tree model was more effective in dealing with the non-linear relationships among the features of network traffic, and its hierarchical decision structure made it possible for the model to make more flexible rules of classification than the linear model, thus increasing the detection of both legitimate and malicious network traffic. Further improvement was achieved by applying the Random Forest ensemble algorithm in which all of the key performance measures exceeded 98% with accuracy of 98.9%. The results indicated that the ensemble of independent decision tree models made the model more robust to the training data noise and variations. Different classifiers predictions were combined, which reduced the variance of the model and therefore, led to a more stable performance on the test data. The most notable was the recall at 98.5%, which demonstrated a significant reduction in missed

attacks compared to the baseline algorithms. The best results were obtained by XGBoost, which got the highest values for all the metrics studied. It reached an accuracy of 99.2%, a precision of 99%, a recall of 98.9%, an F1-score of 98.9%, and a ROC-AUC of 99.5%. The results showed that there was almost complete separation between legitimate and malicious network traffic. This performance was the result of the gradientboosting mechanism, which successively corrected the errors from the previous training iterations and therefore created complex non-linear decision models. The results showed that XGBoost is capable of handling high-dimensional data which involves a large number of interdependent features and complex statistical relationships.

On closer inspection of the ROC-AUC values, it was found that all the models were able to discriminate well. For each of the algorithms, the ROC-AUC value was above 95%, indicating that they could effectively tell the difference between legitimate and malicious traffic at various decision thresholds. However, the ensemble models performed better than the baseline algorithms. The ROC-AUC values for Random Forest and XGBoost were almost 1, meaning that almost perfect discrimination had been achieved between the two types of traffic under examination. On comparing precision and recall, it was found that both the number of false positives and false negatives decreased as the complexity of the algorithmic architecture increased. The improvement in recall was especially significant since this figure indicates how well the system is at identifying actual cyber threats. Recall increased from 92.1% in the case of Logistic Regression to 98.9% with XGBoost, showing that the risk of failing to detect an attack was greatly lowered when the network intrusion detection system was in operation. The experimental results thus showed a clear correlation between the complexity of the algorithmic architecture and the efficiency of cyber threat detection in network traffic. XGBoost achieved the highest score in all the considered criteria and the best trade-off between classification accuracy, attack detection recall and resistance to misclassification. These findings justified the use of XGBoost as the most effective algorithm for the development of modern intelligent network intrusion detection systems intended to analyse network traffic in real time.

Analysis of model confusion matrices. The confusion matrices of all the machine learning models were analysed to determine the nature and structure of misclassifications in network traffic cyber threat detection. Malicious traffic was considered to be the positive class and legitimate network connections the negative class. For each algorithm, the number of true positives, true negatives, false positives and false negatives was calculated. It thus became possible to assess not only the models' overall performance but also the patterns behind their incorrect predictions. The logistic regression results showed that this model had the highest number of false-negative classifications of all the algorithms examined. Because of its linear structure and since it could not model the complex non-linear relationships among the network features, the system was not always able to correctly identify malicious traffic. The model also caused false positives whereby legitimate connections were wrongly classified as attacks. This kind of error showed that there was a greater chance of threats going undetected when the model was used as part of network protection. The Decision Tree algorithm improved the Logistic Regression results since it reduced both types of error. Owing to its capability of considering the non-linear relationships between network traffic parameters and of constructing classification rules through the successive partitioning of the feature space. Nevertheless, the structure of each individual tree still relied on the specific characteristics of the training data, and errors in classification might take place if malicious traffic and normal traffic were similar in their features.

Random Forest committed fewer errors than the previous models. The decrease in false negatives showed that its ability to detect attacks had improved, while the smaller number of false positives meant that normal network connections were identified more accurately. This level of performance was due to the use of an ensemble of Decision Trees, with the final prediction being based on the results of a number of separate classifiers. As a result, the model became more stable, and the effect of the random features in the data was reduced. From the confusion matrix analysis, it was observed that XGBoost achieved the best classification performance as it presented the smallest number of false-positive and false-negative predictions among all models studied. This result indicates that, despite some overlap in the characteristics of malicious and legitimate traffic, XGBoost can properly distinguish between the two. Particularly significant was the reduction in false negatives since such errors could result in actual cyber threats going undetected while the security systems are active. The quantitative results from the confusion matrix analysis are given in Table 2.

Table 2. Confusion matrices of the models examined

Model	True positive	True negative	False positive	False negative
Logistic Regression	36,614	38,152	2,148	3,086
Decision Tree	38,154	39,018	1,282	1,546
Random Forest	39,104	39,812	488	596
XGBoost	39,263	39,901	399	437

Source: compiled by the author

The analysis of the confusion matrix revealed a steady improvement in model performance, since the classification errors decreased as the analysis moved from Logistic Regression through to the tree-based and ensemble algorithms. This trend showed that the models were becoming better at taking into account the complex non-linear relationships which are present in network traffic features. Logistic Regression resulted in the highest number of missed attacks, showing that it had a limited ability to detect the complex multidimensional relationships among the features. The Decision Tree offered a more even distribution of errors since it was able to create non-linear classification rules; it still, however, reacted sensitively to cases in which normal and malicious traffic behaved similarly. The number of false-positive and false-negative results was further reduced by means of Random Forest. The improvement in classification performance was due to the ensemble principle of the algorithm, according to which the final prediction is arrived at by using a number of individual Decision Trees. As a result, the model became more stable, and the impact of specific features of the training data was diminished. XGBoost had the most favourable error profile, giving the smallest number of false alarms and the fewest missed attacks of all the models looked at. This finding verified the algorithm's ability to distinguish between classes even when some of the characteristics of network traffic overlapped.

When considering the errors as a whole, it was discovered that greater algorithmic complexity was associated with fewer misclassifications. The most significant changes were seen in the case of false negatives, as this indicates how effectively the system can detect actual attacks. The results showed that simple linear models are more likely to fail in identifying threats, whereas tree-based and ensemble methods provide a more even distribution of errors. It therefore follows that ensemble algorithms, in particular Random Forest and XGBoost, should be used in systems designed for detecting cyber threats in network traffic, since avoiding missed attacks is a major performance requirement.

Comparison of performance in detecting different attack types. After the general classification performance of the models and their error patterns had been evaluated, a more in-depth analysis was carried out in order to examine the models' capability of recognising the various types of cyber threats found in the CIC-IDS2017 dataset. Different types of attacks produce different network traffic patterns, and therefore the problem of automatic detection of attacks can be quite different. We evaluated the detection performance for each attack category by using the recall and the F1-score as metrics. The recall was quantified as the ability of the model to correctly predict the attack of a specific type among all the cases that actually belong to that category. The F1-score was used as an overall metric of the model's classification performance in terms of the balance between precision and recall, as this is important when the dataset has an imbalanced distribution of the network threat categories. The findings showed that the model's performance depended on the particular characteristics of the attack type, performing best in the case of large-scale denial-of-service attacks and network scanning, as in those situations there were more distinct patterns in the network traffic. On the contrary, web-layer attacks and certain types of brute-force attacks were more difficult to classify since their features resembled normal network traffic. In order to examine the model's robustness in greater detail, the detection of malicious traffic was assessed separately for each of the original attack types found in the source dataset. The recall and F1-score values obtained in this way demonstrated the effectiveness of the binary Attack-class detection for each attack category (Table 3).

Table 3. Performance of binary attack-class detection for the original attack types in the CICIDS2017 dataset using XGBoost

Attack type	Recall, %	F1-score, %
DoS	99.6	99.5
DDoS	99.8	99.7
Port Scan	99.4	99.3
Brute Force	97.8	97.6
Botnet	98.7	98.5
Web Attacks	96.9	96.5

Source: compiled by the author

The analysis carried out on the DoS classification showed that XGBoost was able to correctly identify 99.6% of the cases in this category. This good performance was the result of the clear statistical differences between the network traffic generated during denial-of-service attacks and that which occurs under normal network conditions. Denial-of-service attacks are characterised by a high rate of packet transmission, a sharp rise in the number of simultaneous connections, and changes in network flow parameters, all of which act as obvious signs for automated classification. Success was highest in the case of DDoS attacks, with recall reaching 99.8% and the F1-score amounting to 99.7%, which means that very few of the attacks in this category escaped detection. Even though these attacks are distributed in nature, the general characteristics of the

network traffic showed clear and distinct anomalous patterns that enabled the XGBoost ensemble model to identify them. Furthermore, the classification of port scan attacks reached a recall of 99.4%, demonstrating a high degree of classification performance. This showed the model's ability to detect network reconnaissance, since this type of activity involves a large number of short-lived connections to different destination ports. The behaviour of this kind produces a particular network flow profile which can be effectively differentiated from normal activity.

With regard to Brute Force attacks, lower values were achieved. The recall amounted to 97.8%, and the F1-score was 97.6%. The drop in classification performance was caused by the similarity between some of the stages of these attacks and normal user behaviour. In particular, repeated authentication attempts do not always result in sufficiently obvious anomalies in network traffic, especially when the attack activity is of low intensity. For the Botnet category, the recall was 98.7%, and the F1-score was 98.5%. Although it is hard to detect botnet activity since it can imitate normal network communications, XGBoost was very successful in identifying the typical behaviour patterns of compromised devices. This result supported the notion of using a number of network features in order to detect covert malicious activity. The hardest of all the categories to classify was that of Web Attacks. Of the different kinds of threats examined, this one achieved the lowest recall and F1-score, these being 96.9% and 96.5%, respectively. The reason for these lower scores was that web attacks are generally carried out using the standard HTTP and HTTPS protocols and can display characteristics similar to those of normal user behaviour. It was thus more difficult to distinguish between malicious and legitimate traffic, even when using models that were capable of good classification. A model performance map was made to illustrate the differences in the detection of the various attack categories (Fig. 1).

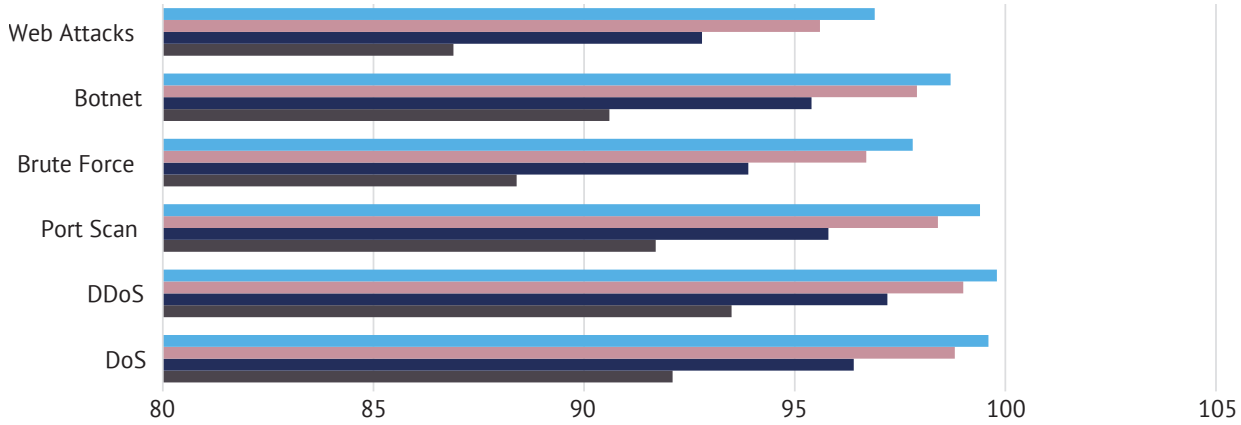


Figure 1. Model performance map by attack class

Source: compiled by the author

The performance map showed a consistent link between the complexity of the algorithmic architecture and the quality of recognition for the various kinds of network attacks. In all the threat categories, Logistic Regression obtained the lowest classification scores, while the XGBoost ensemble model always produced the best results. The greatest improvements in performance were seen in the more complex attack categories, especially Brute Force and Web Attacks. This suggests that standardised ensemble methods are better suited for dealing with network traffic where the signs of malicious activity are not very clear. In general, the performance of the models was directly dependent on the characteristics of each type of malicious traffic. The detection rates of DoS, DDoS and Port Scan attacks were highest, because they caused significant changes in the structure and intensity of network flows. On the other hand, Brute Force, Botnet, and particularly Web Attacks were more difficult to classify automatically because their characteristics were more similar to those of normal network activity. Yet, XGBoost was still able to achieve good performance even in the most difficult threat categories, with recall exceeding 96%. This proves that ensemble algorithms are effective for the design of modern intrusion detection systems intended to analyse network traffic.

Analysis of network traffic feature importance. The analysis indicated that the various network traffic characteristics contributed differently to the process of classification. Only a small number of the features supplied most of the information needed in order to successfully distinguish malicious traffic from genuine network connections. A great many of the other characteristics had a supportive function and were mainly employed to improve the models' classification decisions. The network traffic features which were found to be most important by the feature importance analysis are given in Table 4.

Table 4. Most important network traffic features identified by the feature importance analysis

Feature	Random Forest, %	XGBoost, %
Flow Bytes/s	14.8	16.3
Flow Duration	13.5	14.9
Total Length of Fwd Packets	11.6	12.8
Packet Length Mean	10.4	11.1
Destination Port	8.9	9.6
Flow Packets/s	7.5	8.1
Average Packet Size	6.9	7.3
Fwd Packet Length Mean	5.8	6.2
Bwd Packet Length Mean	5.4	5.1
Flow IAT Mean	4.4	4
Other features	10.8	4.6
Total	100	100

Source: compiled by the author

The findings indicated that the feature importance values obtained by both Random Forest and XGBoost were very similar. Flow Bytes/s had the most significant effect on the classification and referred to the rate of data exchange in a network flow. In the case of many attack types, such as DoS and DDoS, an increase in the amount of transmitted data was one of the main signs of abnormal activity. The high level of importance attributed to this feature thus proved that it played a decisive role in the models' classification decisions. Flow Duration was the second most important feature and stood for the lifetime of a network flow. The temporal characteristics of connections had a significant effect on the identification of different types of network attacks. For instance, port scan attacks usually produced a large number of short-lived connections, while certain kinds of botnet activity involved long-lasting network sessions. This is the reason why Flow Duration ranked among the most influential features.

The total length of forward packets was also very informative since it showed the total amount of data transmitted in the forward direction of a network flow. The significance of this measure was due to the particular data transmission patterns caused by various attacks, patterns which were different from those of normal user traffic. Likewise, the mean packet length served a similar role by indicating the average length of the packets in a flow and by aiding in the detection of statistical differences between normal and malicious network connections. The destination port also merited close attention since it was one of the five most important features for both models. The fact that it was so important showed that information regarding destination network ports was still a key element in detecting possibly harmful activity. This was because a great many network attacks are directed at specific services and network resources, and the analysis of port distributions therefore gave useful information for classification. When a comparative analysis of the results from Random Forest and XGBoost was carried out, it was found that both algorithms had identified almost the same set of the most informative network traffic characteristics. Nevertheless, XGBoost gave a slightly higher share of the total importance to the key features, which was in agreement with its better classification performance. This showed that the algorithm made more effective use of the most important network traffic parameters when drawing the decision boundaries between the different classes. A chart showing the feature importance was prepared in order to present the results visually (Fig. 2).

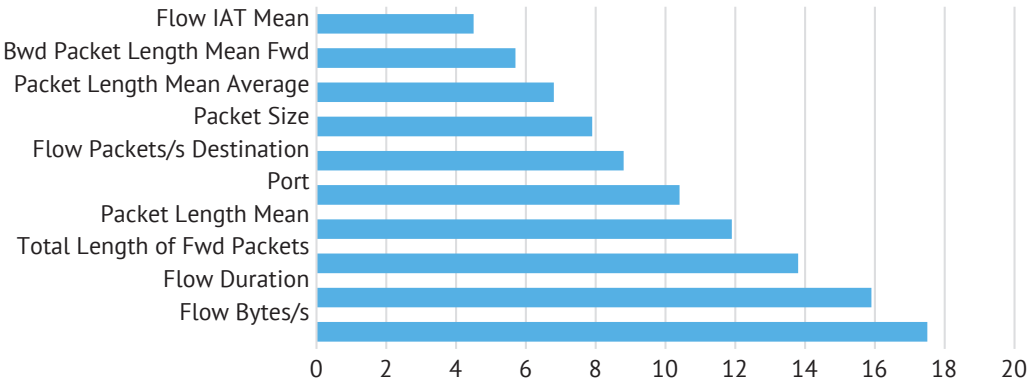


Figure 2. Network Traffic Feature Importance Chart for XGBoost

Source: compiled by the author

The analysis of the chart showed that the most significant factors influencing the identification of a cyber threat were the data flow rate, the time parameters of the network flows and the packet exchange structure. These measures provided the most complete representation of the behaviour characteristics of malicious traffic, and thus allowed the establishment of stable classification criteria. Features that were concerned with the individual, low-level characteristics of network packets, in contrast, were of secondary importance and contributed much less to the classification decisions. The results of the experiment provided a comprehensive insight into the effectiveness of machine learning algorithms in detecting cyber dangers in network traffic using the CIC-IDS2017 dataset. The study investigated the overall classification performance, examined the error patterns of the models, assessed the effectiveness of identifying individual attack types, and explored the effect of certain features on the algorithms' decision-making process. Overall, the results confirmed a direct correlation between the complexity of the model and the quality of detection of malicious network activity.

The results show that Logistic Regression has the lowest scores in the main evaluation metrics. The model was able to achieve an acceptable overall classification accuracy, but it was not able to capture the complex non-linear relationships and the underlying patterns present in the network traffic. Decision Tree, in contrast, obtained a significant improvement in classification performance as it was able to generate non-linear decision rules and could adapt to the structure of the input features. However, the method was sensitive to the features of the training data and was prone to local classification errors. The ensemble algorithms, Random Forest and XGBoost, achieved a significant improvement over the baseline methods. Random Forest reduced both false positives and false negatives by combining a number of independent decision tree models and then combining the predictions that they made. The variance of the model was reduced and its ability to generalise to the test data was increased by this procedure. The best results were achieved by XGBoost which reached the highest values of accuracy, precision, recall and ROC-AUC. This demonstrated its effectiveness in detecting cyber risks in complex input data.

The confusion matrix analysis showed that the ensemble methods significantly reduced the false positive and false negative classifications. In practice, the reduction of false negatives was particularly important, as such errors involve the risk of real attacks going undetected, and cause cybersecurity systems to delay their response. When each type of attack was examined in isolation, the models performed best at identifying DoS, DDoS and Port Scan attacks, whereas Web Attacks and Brute Force attacks remained the most challenging to correctly identify due to their similar network features to those of normal activity. According to the features importance analysis, classification was mainly driven by the features related to the intensity and structure of network traffic, such as Flow Bytes/s, Flow Duration, Total Length of Fwd Packets, Packet Length Mean. The high importance of these features supported the idea that the aggregated statistical parameters of network flows are the most useful for the online danger detection. In contrast, the low-level features of the network packets played a secondary role and had a low impact on the final classification results.

The assessment of the models' strengths and limitations showed that Logistic Regression was performing well due to its fast computational speed and easy interpretability, but it failed to grasp the complex non-linear relationships present in the data. Decision Tree was found more suitable for the network traffic characteristics but it can become unstable if the properties of the sample change. The Random Forest and XGBoost ensemble methods performed very well in classification, and were more robust to noise and class imbalance, but they required more computational resources and longer training times. From a practical point of view, the results obtained have a direct impact on the development of existing intrusion detection and prevention systems (IDS/IPS). Ensemble methods, specifically XGBoost, have improved the ability to automatically detect cyber threats in real time, decreased the number of attacks that were missed, and reduced the number of false alarms. Furthermore, the set of most informative features identified can be used to optimise computational costs by reducing the dimension of the input space without greatly affecting the classification performance.

DISCUSSION

Other researchers generally agree with the current results as to the effectiveness of machine learning techniques, especially ensemble methods, in the area of network intrusion detection. S. Farhat *et al.* (2023) found that classical machine learning algorithms were able to achieve high classification accuracy in the detection of DoS and DDoS attacks using the CIC-IDS2017 dataset, whereas ensemble methods proved to be more robust in terms of false alarms. This is in line with the present study, since XGBoost obtained the highest values for accuracy, recall, and the F1-score together with the lowest error rate. D. Akgun *et al.* (2022) suggested a deep learning model for the detection of DDoS attacks, which was more capable than classical algorithms of identifying complex non-linear patterns in network traffic. The authors highlighted the benefits of using complex models for the purpose of anomaly detection. This is in line with the performance trend seen in the current study, going from Logistic Regression to XGBoost, the latter showing the best generalisation ability. On the other hand, L. D'hooge *et al.* (2023) discovered that even

highly accurate machine learning models can suffer a significant drop in performance when there is a change in the data distribution or when they are used on new datasets that are only weakly related. For this study, the reason for the poorer classification results with respect to the more complicated attack types – namely Web Attacks and Brute Force – is that the features of these attacks overlapped with those of normal traffic. However, ensemble methods, especially XGBoost, lessened the seriousness of this problem because of their greater generalisation capacity.

Further investigations showed a close similarity between the present results and the general trends in the development of network intrusion detection systems. N. Thapa *et al.* (2020), F. Genuario *et al.* (2024) and Z.I. Khan *et al.* (2024) found that the performance of cyber threat detection depended directly on the models' capacity to consider complex non-linear patterns in network traffic, with ensemble machine learning algorithms being the most effective. In particular, N. Thapa *et al.* (2020) found that machine learning models greatly surpassed rule-based methods since they adapted better to complex data structures and led to fewer false negatives. An identical effect was observed in the present study, since the use of ensemble methods greatly decreased the number of attacks that were missed. Z.I. Khan *et al.* (2024) also pointed out that Brute Force and Web Attacks remained among the most difficult types to detect because they are similar to normal user activity. The findings given here fully agree with this conclusion: although the models as a whole performed very well, the different attack types had the lowest recall and F1-score values. This implies that the main limitation of the current algorithms is not their overall classification accuracy but rather the difficulty of distinguishing malicious traffic from legitimate traffic when the anomalies are subtle. To summarise, the results from the comparison show that the improvements in the intrusion detection performance depend on two closely related factors, i.e., the use of more powerful ensemble algorithms and the availability of informative network traffic features.

M. Pujari *et al.* (2022), M. Cantone *et al.* (2024), and M. Sajid *et al.* (2024) have shown that in the research of network intrusion detection, in addition to improving the classification accuracy, the robustness of the model in view of changing operating conditions and complex attack scenarios has been the focus. The team headed by M. Cantone *et al.* (2024) discovered that a high degree of accuracy on one dataset did not guarantee similar performance when the model was applied to other sources of network traffic. They found that changes in the data structure and in the characteristics of network activity led to a decline in classification performance. This result is consistent with those of the present study, which also found that the model's performance depended on the particular type of attack. M. Pujari *et al.* (2022) demonstrated that machine learning models can become less effective when exposed to data that has been deliberately modified or is atypical. It was found that making small changes to the features could result in a rise in the number of classification errors. This helps to account for the poorer detection of Brute Force and Web Attacks in the current study, as these categories exhibited subtle anomalies and had a large amount of feature overlap with normal network activity. M. Sajid *et al.* (2024) observed that hybrid architectures, which combine different machine learning and deep learning techniques, attain a higher degree of accuracy than individual models do. They explained this advantage by stating that hybrid methods are better at capturing complex non-linear patterns in network traffic. A similar trend was noted in this study: as the complexity of the algorithmic architecture increased, so did the classification performance, with XGBoost providing the best results.

Various studies have looked into hybrid, ensemble, and multi-model methods aimed at enhancing the accuracy of network attack detection. An examination of the research carried out by A. Kumar & D. Pandey (2024), V. Sharma & M. Kumar (2025) and G. Rajesh *et al.* (2026) found that combining different algorithms or architectures led to better classification performance and a decrease in misclassifications when compared to using individual models. This is in agreement with the current study, since ensemble methods also resulted in the best performance. Specifically, XGBoost achieved the highest accuracy and the lowest number of missed attacks. The research carried out by G. Rajesh *et al.* (2026) and by V. Sharma & M. Kumar (2025) also highlighted the issue of model generalisation across various datasets. In the current study, a similar difficulty was shown by the inadequate classification of Brute Force and Web Attacks, since these types of attacks most closely resembled normal traffic. Furthermore, the results obtained by A.F. Al-zubidi *et al.* (2024) confirmed that DoS and DDoS attacks are among the most easily detected kinds of cyber threats since they cause obvious network traffic anomalies. The same trend was found in this study, with these attack categories reaching the highest recall and F1-score values. Altogether, the findings backed the present tendency to use ensemble and hybrid models in network intrusion detection.

D. Sudyana *et al.* (2024) looked at how well machine learning models could generalise across different datasets when it came to detecting network attacks and discovered that their performance dropped significantly when the models were transferred from one dataset to another. They regarded changes in feature distributions and in the overall traffic structure as the main reasons for the decline in classification performance. This is in close agreement with the current results, since complex types of attack – especially Web Attacks and Brute Force – performed worse in terms of classification

because their features overlapped with those of normal traffic. H. Hakami *et al*. (2025) studied the application of machine learning techniques to enhance attack detection in Internet of Things environments and found that ensemble algorithms surpassed the baseline models, especially in situations involving limited resources and highly variable traffic. The authors also stressed the importance of choosing informative features in order to lower computational demands, something which matches the current finding that aggregated network traffic characteristics were central to the issue.

Researchers have likewise paid attention to the detection of web attacks, feature optimisation, and the real-time use of ensemble models. M.K. Baklizi *et al*. (2024) presented a machine-learning-based system for the detection of web attacks and discovered that algorithms able to take into account the non-linear relationships between network features obtained the best results. The authors pointed out how difficult it is to identify malicious web traffic since it is similar to normal activity. This finding is entirely in agreement with the current results, since Web Attacks also had the lowest classification performance. A. Bilal *et al*. (2026) suggested a hybrid feature-selection framework that involved the use of ensemble models in order to enhance intrusion detection. They discovered that combining the selection of informative features with ensemble algorithms greatly reduced classification errors and improved the model's generalisation. This is in agreement with the current results, since the aggregated network traffic features were of key importance and XGBoost offered the best compromise between accuracy and robustness. I. Hamdaoui *et al*. (2024) assessed various machine learning algorithms for the purpose of intrusion detection in high-volume real-time data streams. It was found that ensemble methods gave the best balance between accuracy and processing speed, a feature that is essential for real-time systems. This is exactly what the present findings show, since XGBoost and Random Forest achieved the best performance while at the same time minimising critical errors, especially false negatives.

Machine learning models have similarly shown excellent performance when it comes to classifying DDoS attacks and flash-event network anomalies, attaining high levels of both accuracy and recall. According to U. Nag *et al*. (2025), ensemble methods yielded the most stable results since they were more effective at dealing with traffic variability and data noise. This is in line with the current study, in which ensemble models, especially XGBoost, achieved the best detection performance for DDoS and DoS attacks and resulted in the fewest false-negative predictions. K. Ren *et al*. (2022) put forward a deep reinforcement learning method aimed at optimising feature selection for use in intrusion detection. They showed that automated feature-selection techniques could considerably enhance model performance by reducing the dimension of the feature space and getting rid of redundant information. Together with the current results – where aggregated statistical features of network traffic, for example Flow Bytes/s and Flow Duration, were of central importance – these findings prove that it is essential to select the right informative features if strong classification performance is to be achieved. In general, recent studies have verified the main trends found in this work: a decline in performance when there is a distribution shift, vulnerability to complex or adversarial situations, and the advantage of ensemble and hybrid models. The results also indicated that combining informative features with non-linear ensemble algorithms led to the highest level of robustness and accuracy in the area of network intrusion detection.

CONCLUSIONS

The experimental study found a direct link between the complexity of the algorithms used in the models and the effectiveness of cyber threat detection in network traffic. All of the algorithms looked at obtained high scores for the main evaluation criteria. But when they were compared, it was seen that performance steadily increased as one went from simple models to ensemble methods. Thus, it was found that the increase in the algorithmic complexity improved the models' ability to correctly classify network connections. Logistic Regression had the lowest values among all models examined with an accuracy of 94.7%, a recall of 92.1%, and an F1-score of 92.8%. These results showed that the linear algorithm was unable to learn the complex non-linear relationships existing in the high-dimensional feature space of network traffic. The relatively low recall also showed that more number of attacks were missed which is a serious concern in the intrusion detection systems.

The performance of the classification was improved by applying the Decision Tree algorithm, with the accuracy reaching 97.2% and the recall reaching 96.1%. The better performance could be due to the model's ability to create non-linear rules for partitioning the data, and to consider the complex relationships that exist between the different features. However, the algorithm remained sensitive to changes in the data structure that could in turn influence the stability of its predictions in difficult cases. The Random Forest ensemble algorithm was used to achieve further performance improvement. Recall was 98.5% and the F1-score was 98.6%, which indicates a marked reduction in classification error. This improvement was achieved by using a number of decision tree models, and combining their outputs to give the final prediction. The method reduced the impact of errors of the individual base-model and improved the stability of the prediction. XGBoost was the most effective of all the models examined, achieving an accuracy of 99.2%, a recall of 98.9%, an F1-score of 98.9%, and

a ROC–AUC of 99.5%. It made the fewest false-positive and false-negative predictions. The number of false negatives reduced from 3,086 in the case of Logistic Regression to 437 with XGBoost and the number of false positives reduced from 2,148 to 399. These results showed the model had better ability of accurately distinguishing legitimate and malicious network traffic.

XGBoost was found to be the most effective for detecting DoS, DDoS and Port Scan attacks, with a recall rate of up to 99.8%. Brute Force, and Botnet and Web Attacks were harder to detect because they are similar to a normal network traffic. However, the recall rate for these types of attacks was still higher than 96%. The feature importance analysis revealed that the features which contributed the most to the classification were Flow Bytes/s, Flow Duration, Total Length of Fwd Packets, Packet Length Mean and Destination Port. The more complicated the algorithm architecture, the better the classification performance and the lower the error rate, especially false negative rate. XGBoost is the best result for the use of ensemble models and the aggregation of network traffic features in modern intrusion detection and prevention systems (IDS/IPS). Further work is needed to improve the proposed method through the addition of deep learning techniques, real-time stream processing, and adaptive mechanisms for the detection of the new types of network attacks.

ACKNOWLEDGEMENTS

None.

FUNDING

None.

CONFLICT OF INTEREST

None

REFERENCES

[1] Ahmed, N., Saleem, G., & Naveed, A. (2026). A resource-efficient machine learning pipeline for DDOS attack detection: A comparative study on CIC-IDS2018 and CIC-DDoS2019. *ICCK Transactions on Information Security and Cryptography*, 2(1), 55-69. doi: 10.62762/TISC.2025.438083.

[2] Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Computers & Security*, 118, article number 102748. doi: 10.1016/j.cose.2022.102748.

[3] Alhakeem, M.S., & Ajlan, K.B. (2024). A comparative evaluation of machine learning-based intrusion detection systems for securing cloud environments. *Journal of Information Security and Cybercrimes Research*, 7(2), 127-142. doi: 10.26735/RSND3740.

[4] Al-zubidi, A.F., Farhan, A.K., & Towfek, S.M. (2024). Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model. *Journal of Intelligent Systems*, 33(1), article number 20230195. doi: 10.1515/jisys-2023-0195.

[5] Baklizi, M.K., Atoum, I., Alkhazaleh, M., Kanaker, H., Abdullah, N., Al-Wesabi, O.A., & Otoom, A.A. (2024). Web attack intrusion detection system using machine learning techniques. *International Journal of Online & Biomedical Engineering*, 20(3), 24-38. doi: 10.3991/ijoe.v20i03.45249.

[6] Bilal, A., Sharif, K., Zhu, L., Li, F., Xu, C., & Karim, M.M. (2026). Evaluation to integration: Hybrid feature selection framework with ensemble machine learning for intrusion detection. *IEEE Transactions on Dependable and Secure Computing*, 23(3), 6362-6377. doi: 10.1109/TDSC.2026.3664110.

[7] Cantone, M., Marrocco, C., & Bria, A. (2024). Machine learning in network intrusion detection: A cross-dataset generalization study. *IEEE Access*, 12, 144489-144508. doi: 10.1109/ACCESS.2024.3472907.

[8] Chua, T.H., & Salam, I. (2023). Evaluation of machine learning algorithms in network-based intrusion detection using progressive dataset. *Symmetry*, 15(6), article number 1251. doi: 10.3390/sym15061251.

[9] Coşar, H.İ., Arısoy, Ç., & Ulutaş, H. (2024). Intrusion detection on CSE-CIC-IDS2018 dataset using machine learning methods. *Artificial Intelligence Theory and Applications*, 4(2), 143-154.

[10] D'hooge, L., Verkerken, M., Wauters, T., De Turck, F., & Volckaert, B. (2023). Investigating generalized performance of data-constrained supervised machine learning models on novel, related samples in intrusion detection. *Sensors*, 23(4), article number 1846. doi: 10.3390/s23041846.

[11] Elhanashi, A., Gasmi, K., Begni, A., Dini, P., Zheng, Q., & Saponara, S. (2022). Machine learning techniques for anomaly-based detection system on CSE-CIC-IDS2018 dataset. In R. Berta & A. De Gloria (Eds.), *International conference on applications in electronics pervading industry, environment and society* (pp. 131-140). Cham: Springer. doi: 10.1007/978-3-031-30333-3_17.

[12] Farhat, S., Abdelkader, M., Meddeb-Makhlouf, A., & Zarai, F. (2023). Evaluation of DoS/DDoS attack detection with ML techniques on CIC-IDS2017 dataset. In *Proceedings of the 9th international conference on information systems security and privacy* (pp. 287-295). Lisbon: SciTePress. doi: 10.5220/0011605700003405.

[13] Genuario, F., Santoro, G., Giliberti, M., Bello, S., Zazzera, E., & Impedovo, D. (2024). Machine learning-based methodologies for cyber-attacks and network traffic monitoring: A review and insights. *Information*, 15(11), article number 741. doi: 10.3390/info15110741.

[14] Hakami, H., Faheem, M., & Ahmad, M.B. (2025). Machine learning techniques for enhanced intrusion detection in IoT security. *IEEE Access*, 13, 31140-31158. doi: 10.1109/ACCESS.2025.3542227.
[15] Hamdaoui, I., El Makkaoui, K., & El Allali, Z. (2024). Evaluating machine learning algorithms for intrusion detection: A step toward securing real-time Big Data. In *Proceedings of the 7th international conference on advanced communication technologies and networking* (pp. 1-7). Rabat: IEEE. doi: 10.1109/CommNet63022.2024.10793323.
[16] Ilyas, M.U., & Alharbi, S.A. (2022). Machine learning approaches to network intrusion detection for contemporary internet traffic. *Computing*, 104, 1061-1076. doi: 10.1007/s00607-021-01050-5.
[17] Khan, Z.I., Afzal, M.M., & Shamsi, K.N. (2024). A comprehensive study on CIC-IDS2017 dataset for intrusion detection systems. *International Research Journal on Advanced Engineering Hub*, 2(2), 254-260. doi: 10.47392/IRJAEH.2024.0041.
[18] Kumar, A., & Pandey, D. (2024). Enhancing intrusion detection with ML and deep learning: A survey of CICIDS 2017 and CSE-CIC-IDS2018 datasets. *AIP Conference Proceedings*, 3168, article number 020003. doi: 10.1063/5.0222131.
[19] Nag, U., Patil, N.V., Bhardwaj, A., & Krishna, C.R. (2025). Classification of DDoS attacks and flash events using machine learning: Comparative evaluation on CIC and simulated datasets. In *Proceedings of the 2nd global AI summit-international conference on artificial intelligence and emerging technology (AI Summit)* (pp. 54-59). Noida: IEEE. doi: 10.1109/AISummit66170.2025.11411154.
[20] Nikitenko, A., & Sobol, Y. (2025). Inreractive system for detecting network intrusions in real time based on deep learning. *Bulletin of Lviv State University of Life Safety*, 32, 100-110. doi: 10.32447/20784643.32.2025.08.
[21] Pujari, M., Pacheco, Y., Cherukuri, B., & Sun, W. (2022). A comparative study on the impact of adversarial machine learning attacks on contemporary intrusion detection datasets. *SN Computer Science*, 3, article number 412. doi: 10.1007/s42979-022-01321-8.
[22] Rajesh, G., Prathyusha, G., & Dheeraj, A. (2026). A multi-model deep learning framework for enhanced network intrusion detection across diverse datasets. *American Journal of AI Cyber Computing Management*, 6(2), 332-341. doi: 10.64751/ajaccm.2026.v6.i2.410.
[23] Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports*, 12, article number 15370. doi: 10.1038/s41598-022-19366-3.
[24] Rodriguez, M., Alesanco, A., Mehavilla, L., & García, J. (2022). Evaluation of machine learning techniques for traffic flow-based intrusion detection. *Sensors*, 22(23), article number 9326. doi: 10.3390/s22239326.
[25] Saini, N., Bhat Kasaragod, V., Prakasha, K., & Das, A.K. (2023). A hybrid ensemble machine learning model for detecting APT attacks based on network behavior anomaly detection. *Concurrency and Computation: Practice and Experience*, 35(28), article number e7865. doi: 10.1002/cpe.7865.
[26] Sajid, M., Malik, K.R., Almogren, A., Malik, T.S., Khan, A.H., Tanveer, J., & Rehman, A.U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing*, 13, article number 123. doi: 10.1186/s13677-024-00685-x.
[27] Sharma, V., & Kumar, M. (2025). Improving intrusion detection with hybrid deep learning models: A study on CIC-IDS2017, UNSW-NB15, and KDD CUP 99. *Journal of Information Systems Engineering and Management*, 10(11s), 633-650. doi: 10.52783/jisem.v10i11s.1665.
[28] Storozenko, V., Demianchuk, B., Kozlovska, D., & Fesenko, A. (2025). Overview of IDS capabilities for network traffic analysis. *Science-Based Technologies*, 67(3), 317-324. doi: 10.18372/2310-5461.67.20036.
[29] Sudyana, D., Lin, Y.-D., Verkerken, M., Hwang, R.-H., Lai, Y.-C., D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2024). Improving generalization of ML-based IDS with lifecycle-based dataset, auto-learning features, and deep learning. *IEEE Transactions on Machine Learning in Communications and Networking*, 2, 645-662. doi: 10.1109/tmlcn.2024.3402158.
[30] Thapa, N., Liu, Z., KC, D.B., Gokaraju, B., & Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet*, 12(10), article number 167. doi: 10.3390/fi12100167.

Експериментальна оцінка моделей машинного навчання для виявлення кіберзагроз у мережевому трафіку

Ігор Послєд
Аспірант
Національний технічний університет «Дніпровська політехніка»
49005, просп. Дмитра Яворницького, 19, м. Дніпро, Україна
https://orcid.org/0009-0006-0885-2007

Анотація. Метою цього дослідження було підвищення здатності автоматизованих систем виявляти кіберзагрози в мережевому трафіку шляхом використання та порівняння різних моделей машинного навчання з огляду на зростаючу складність атак та збільшення обсягу мережевих даних. Завданням цього дослідження є визначення характеристик та ефективності стандартних моделей машинного навчання для виявлення кіберзагроз у мережевому трафіку шляхом експериментальної оцінки. Це було експериментальне, кількісне емпіричне дослідження в галузях машинного навчання та кібербезпеки, яке проводилося за допомогою комп'ютера. Дослідження базувалося на одноцикловому обчислювальному експерименті з набором даних CIC-IDS2017, в ході якого моделі були навчені, оцінено їхню ефективність та проаналізовано закономірності в помилках класифікації. Результати показали чіткий взаємозв'язок між алгоритмічною складністю та якістю виявлення кіберзагроз у мережевому трафіку. Усі моделі отримали високі значення за різними показниками, хоча ансамблеві методи завжди демонстрували кращі результати, ніж базові підходи. Логістична регресія показала найгірші результати: точність 94,7 %, відкликання 92,1 % та F1-показник 92,8 %. З іншого боку, XGBoost досяг найкращих результатів: точність 99,2 %, відкликання 98,9 %, F1-показник 98,9 % та площа під кривою ROC 99,5 %, при цьому він також спричинив найменшу кількість помилкових класифікацій. Аналіз матриць плутанини показав значне зменшення кількості пропущених атак – їхня кількість знизилася з 3 086 до 437 при використанні XGBoost, що дозволило зменшити ризик того, що загрози залишаться невиявленими. Найвища ефективність класифікації була зафіксована для атак типу «відмова в обслуговуванні», «розподілена відмова в обслуговуванні» та «сканування портів», де показник відтворюваності досяг 99,8 %. Атаки методом грубої сили, активність ботнетів та веб-атаки було складніше виявити, оскільки вони нагадували легітимний трафік, хоча показник відтворюваності для цих категорій також залишався на рівні понад 96 %. Ці результати можуть бути використані фахівцями з кібербезпеки, розробниками систем виявлення та запобігання вторгненням, а також аналітиками центрів оперативної безпеки

Ключові слова: ансамблеві алгоритми; XGBoost; матриця помилок; значущість ознак; система виявлення (запобігання) вторгнень; розподілена відмова в обслуговуванні; ботнет; CIC-IDS2017