İSTANBUL UNIVERSITY PRESS

Acta Infologica
acin 2024, 8 (1), 34–50
DOI: 10.26650/acin.1444470

Research Article / Araştırma Makalesi

Network Forensics Analysis of Cyber Attacks on Computer Systems using Machine Learning Techniques

Firdevs Yıldız[1], Batuhan Gül[1],Fatih Ertam[1]

[1]Fırat University, Faculty of Technology, Department of Digital Forensics Engineering, Elazığ, Türkiye

Corresponding author : Batuhan Gül
E-mail : b.gul@firat.edu.tr

ABSTRACT

With the rapid development of technology, significant progress has been observed regarding the Internet and interconnected devices, increasing the risk of cyberattacks targeting these platforms. These attacks take diverse and sophisticated forms and pose a serious threat to companies, potentially causing substantial financial losses and service disruptions. In response, the pressing need exists to develop robust defense strategies. This research focuses on analyzing attacks on information systems, specifically concentrating on network forensics using machine learning techniques. The initial phase involves executing various attack scenarios in a virtual environment, recording network packets, and extracting relevant features to create a dataset. A classification framework is then created that includes machine learning algorithms such as random forest, support vector machine (SVM), and Naïve Bayes. Comparing the performance of these algorithms on the study's dataset has revealed the random forest algorithm to achieve the highest accuracy rate at 94.8%, with Naive Bayes having the lowest at 78.9

Keywords: Machine learning, cyberthreat, network forensics, classification algorithms, intrusion detection system

Submitted : 28.02.2024
Revision Requested : 05.04.2024
Last Revision Received : 15.04.2024
Accepted : 09.05.2024
Published Online : 11.06.2024

1. INTRODUCTION

In tandem with the continual advancement of technology, computer networks have become an indispensable component in nearly every facet of human life and offer convenience in numerous domains. Termed as structures that facilitate data exchange and communication among computers, computer networks enable seamless interactions among these devices. They serve various beneficial purposes, including data transmission, information exchange, and resource sharing. The increased prevalence of computer network utilization corresponds with a parallel escalation in the significance of network security (Akbal et al., 2019). Due to their ubiquity and paramount importance in human life, computer networks have become a target for cyberattacks.

Cyberattacks directed toward computer networks have transcended the realm of information operations, exerting a broader impact. They not only pose a threat to individual users but also jeopardize the security of institutions, governments, and overall societal well-being. These attacks manifest in various forms, including unauthorized network access, data theft, service disruption, ransomware incidents, and other malicious activities. In addition to inflicting material damages, cyberattacks can lead to broader consequences such as compromised personal privacy, disclosure of trade secrets, and implications for national security (Li & Liu, 2021). Hence, attacks on computer networks have heightened the need for network forensics. Network forensics is a subcategory of digital forensics that employs scientific methodologies to allows electronic evidence related to a crime to be presented in an understandable and unaltered state before judicial authorities. Digital forensics encompasses the entirety of the evidence examination process and employs scientific techniques to facilitate the elucidation of a crime (Başlar, 2020). Meanwhile, network forensics constitutes a digital forensic process encompassing the investigation, analysis, and monitoring of computer networks (Qureshi et al., 2021). Network forensics involves the monitoring of a network for abnormal traffic and the identification of unauthorized entries. An assailant may expunge all log files from a compromised central computer, rendering network-based evidence the sole available proof for forensic analysis in such circumstances (Hunt, 2012). Hence, with the increasing complexity of attacks, the significance of network forensics has further heightened.

With the increasing number and diversity of cyber threats, the field of security is undergoing rapid development. Numerous software and hardware network security tools such as security firewalls, antivirus programs, and intrusion detection systems have been developed (Özekes & Karakoç, 2019). Machine learning is currently gaining popularity due to how it provides a set of methods and techniques that yield high accuracy for detecting attacks. The advancement of machine learning methods and techniques presents new opportunities in the field of network forensics. Machine learning is a subfield of artificial intelligence (AI) that enables computers to learn from data without being explicitly programmed, thus facilitating the resolution of complex problems (Bi et al., 2019). Machine learning algorithms make predictions about new data based on training data without explicitly specifying how models will be applied. Machine learning techniques are garnering significant attention across various industries. In the field of cybersecurity, these techniques are being applied for detecting new and sophisticated attacks, thus contributing to the advancement of cybersecurity(Shaukat et al., 2020). This study focuses on analyzing attacks on computer systems from the perspective of network forensics using machine learning techniques and addresses fundamental concepts in network forensics, machine learning techniques, and studies related to the application of these techniques in the context of network forensics. The study conducts various attacks in a virtual environment and uses different learning methods to run classification processes based on the dataset generated from these attacks for subsequent analysis.

Briefly, the main contributions of this review can be stated as follows:

- The study generates attack scenarios on computer systems, classifies these attack types using machine learning methods, and compares the performances of the classification algorithms by creating and employing different attack scenarios.
- The study then analyzes the attacks on information systems from the perspective of network forensics using machine learning techniques.
- The study also executes diverse attack scenarios in a virtual environment, capturing network packets from the conducted attacks, extracting features, and subsequently constructing a new dataset.
- The study utilizes the created dataset to perform classifications using the random forest, support vector machine (SVM), Naive Bayes, logistic regression, and decision tree classification algorithms. The classification results are compared using performance metrics. In the experiments, the random forest classifier ranked the highest in performance by scoring 94.8% in accuracy, 98% in precision, 91.8% in recall, and 92.4% in F1 score.

The remainder of this paper is organized as follows. Section 2 provides a summary of previous studies conducted in this field. The third section discusses machine learning and the classification algorithms. Section 4 introduces the proposed method and creates the dataset. Section 5 classifies the machine learning classifiers used in the experiments on the dataset. The final section discusses the findings and future directions.

2. RELATED WORKS

Wani et al. (2019) presented an approach aimed at detecting distributed denial-of-service (DDoS) attacks in cloud computing environments. Their study conducted attacks using the Tor Hammer attack tool in the ownCloud environment and created a dataset. They employed SVM, Naïve Bayes, and random forest machine learning algorithms to detect DDoS attacks and compared the success rates using performance metrics. The results of the experiments indicated the SVM model to exhibit higher accuracy and precision compared to Naïve Bayes and random forest.

İnce et al. (2021) addressed and compared machine learning algorithms. They utilized the NSL-KDD dataset to evaluate the comparisons. They then applied different tests on the NSL-KDD dataset and used 5% of, 10% of, and then the entire dataset for these tests. They calculated the models' performances using accuracy and F-score values, with the overfitting machine learning (OML) algorithm achieving the highest performance at 99.8% accuracy and a 99.9% F-score.

Ahmetoğlu and Das (2021) proposed an approach based on intuitive feature selection and machine learning to detect web application attacks using hybrid intrusion detection systems. They combined the web application attacks and normal flow examples from the CSE-CIC-IDS2018 dataset after the data preprocessing steps to create a new dataset. They then used the created dataset for calculating the average mean square error and feature count optimization using the genetic algorithm and logistic regression. They tested this optimization process with five different machine learning algorithms: random forest, SVM, Naïve Bayes, k-nearest neighbors (KNN), and deep neural networks (DNN). The most successful methods for classification were RF, KNN, and DNN, respectively. Upon examining the results, they observed the number of features to be reduced by 85% while keeping the classification success rates at the 99% level.

Radivilova et al. (2019) examined machine learning classification methods for detecting DDoS attacks. They tested the performances of machine learning algorithms using datasets created with different DDoS attack scenarios. They also evaluated the detection performance of the algorithms using performance metrics such as accuracy rate, precision, specificity, and F1 score. Their evaluation results emphasize the successful application of classification methods for detecting DDoS attacks, particularly the random forest algorithm.

Ferrag et al. (2020) presented a study involving deep learning methods for detecting cyberattacks using datasets such as CSE-CIC-IDS2018 and Bot-IoT and examined deep learning methods including recurrent neural networks (RNN), DNN, restricted Boltzmann machines (RBM), deep belief networks (DBN), convolutional neural networks (CNN), deep Boltzmann machines (DBM), and deep autoencoders (DA). They categorized the literature's 35 attack detection datasets and analyzed the performance of deep learning methods on these datasets.

Dina et al. (2021) investigated the use of machine learning methods for detecting unauthorized entries. They employed various machine learning techniques such as decision trees, SVM, artificial neural networks (ANN), random forest, and Naïve Bayes classifiers for unauthorized entry detection and examined their advantages and disadvantages. Additionally, they utilized datasets created to represent various types of attacks and performed evaluations using performance metrics.

Shafiq et al. (2020) discussed the importance of identifying cyberattack traffic for IoT security. They proposed an effective machine learning algorithm selection framework and a hybrid algorithm for identifying anomalies and unauthorized entries in IoT network traffic. The dataset comprises normal traffic, attacks, and Internet of Things (IoT) Botnet attacks for identifying and detecting the best attacks in the IoT network. They selected and accurately applied five machine learning algorithms (i.e., Naïve Bayes, Bayes Net, C4.5 decision tree, random forest, and random tree). They then evaluated the performance of the applied machine learning algorithms and stated the performance of Naïve Bayes and random tree to be highly effective compared to the other machine learning algorithms. When comparing the performance results from the Naïve Bayes and random tree machine learning algorithms, they indicated Naïve Bayes to be the most effective machine learning algorithm.

Shaukat et al. (2020) presented an evaluation of commonly used machine learning methods for detecting certain cyber threats by examining three fundamental machine learning techniques (i.e., deep belief network, decision tree, and SVM). Based on frequently used datasets considered as references, they briefly evaluated the performance of these machine learning techniques in the areas of unwanted email detection, unauthorized entry detection, and malware detection.

Zhang et al. (2019) proposed a network attack detection method based on a deep hierarchical network and original flow data for the CICIDS2017 and CTU datasets. Their method used CNN classification to learn spatial features and long short-term memory (LSTM) classification to learn temporal features. As a result of the classification process with

CNN+LSTM, they detected data in the CICIDS2017 dataset with 99.8% accuracy and in the CTU dataset with 98.7% accuracy.

Aamir et al. (2021) conducted a study on a subset of the CICIDS2017 dataset, specifically the Juma-working hours-afternoon dataset, which includes Benign, PortScan, and DDoS labels. The first stage of the study removed features with infinite or calculated values from the dataset. Correlations between features were calculated, and features with correlations below 20% were eliminated. They used the remaining features to create a new dataset and then applied standard scaler normalization . Approximately 70% of the obtained dataset was allocated for training, and 30% for testing. Classification algorithms and some ensemble classifiers were used, resulting in accuracy values of 60.6%, 97.1%, 99.0%, 68.7%, and 85.5%. These results provide an analysis of different algorithm performances and variability regarding classification accuracy.

Karaman et al. (2021) used ANN to analyze attack detection models using the CSE-CIC-IDS 2018 dataset. Their study created five separate sub-datasets and selected features for each sub-dataset. They used the created models to determine whether attacks were DDoS, BruteForce, Botnet, or denial-of-service (DoS) attacks and to identify the type of attack, achieving accuracy values of 99.11%, 99.31%, 99.26%, 93.23%, and 92.26% for each sub-dataset.

Aslan and Yilmaz (2021) proposed a deep learning-based architecture capable of classifying malware derivatives effectively using a hybrid model. They first collected data and then designed the DNN to be used in the study. They then tested the proposed model on the Malimg, Microsoft BIG 2015, and Malevis datasets and reached high performance rates compared to previous examples in the literature, with testing on the Malimg dataset achieving an accuracy of 97.78%.

Al-Zubi et al. (2021) presented a model for the security and privacy of patient data. Using machine learning models, this approach predicted cyberattack behavior in the healthcare field and facilitated the processing of this data. The proposed approach is based on a patient-centric design, allowing users to control data sharing access by protecting information on trusted devices such as their mobile phones. Experimental results indicated the proposed model to provide a higher attack prediction rate (96.5%), accuracy rate (98.2%), and efficiency rate (97.8%) with lower latency (21.3%) and reduced communication costs (18.9%) compared to other existing models.

Pallathadka et al. (2023) proposed a method that compares the performance of students in an institution using machine learning methods. They investigated the performance of such machine learning methods as Naïve Bayes, ID3, C4.5, and SVM using the UCI machinery student performance dataset. As a result of the comparison, the SVM algorithm was seen to have the highest accuracy rate.

Suryadevara (2023) proposed a new method to detect whether a person has diabetes or not using machine learning techniques. The study used the diabetes dataset and compared such classification algorithms as KNN, logistic regression, and random forest over this dataset. As a result of comparing five machine learning methods, decision tree was seen to achieve the highest accuracy rate at 99

Ashton et al. (2023) investigated the advantage and potential of using machine learning in the field of pediatrics. As a result, their research predicted machine learning to be able to greatly help people manage pediatric conditions over the next 5-10 years.

Noella and Priyadarshini (2023) proposed a novel machine learning-based method that analyzes Parkinson's and Alzheimer's diseases. They compared the performances of bagged ensemble, ID3, Naïve Bayes, and multiclass support vector machine classifiers using the Positron Emission Tomography (PET) dataset. In the comparison, bagged ensemble showed a higher performance than other machine learning algorithms with an accuracy of 90.3%, sensitivity of 89%, specificity of 92%, and precision of 87%.

The majority of previous studies have been on health issues, with a very limited number of studies observed to have occurred on analyzing the network forensics of attacks on computer systems using machine learning methods in the past years.

3. MACHINE LEARNING

Machine learning is a subfield of AI and had its foundations laid in the 1950s. In 1959, Arthur Samuel coined the term machine learning and conducted studies related to algorithms used in computer games. With the advancements in technology and data science over the years, machine learning has evolved into a significant scientific discipline that possesses the ability to self-learn through algorithms. The data provided to a system is recognized through the employed algorithms, and the system responds accordingly in the output, thus becoming intelligent over time without human intervention (Sharma et al., 2021).

Currently, machine learning is being applied in various fields, and new algorithms and models are continually being developed. Machine learning algorithms have been instrumental at addressing various issues and contributing to the advancement of fields such as cybersecurity and digital forensics. The capabilities of machine learning enable the detection of cyber threats, the creation of predictions, and the identification of anomalies in a network. The prediction steps using machine learning methods are illustrated in Fig. 1.

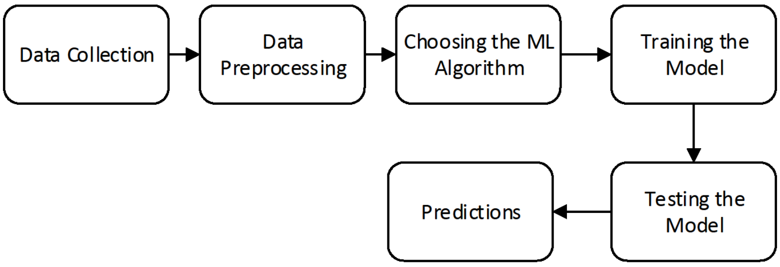


Figure 1. The steps of machine learning.

3.1. Classification Algorithms

This study creates its own dataset to measure the performance of classification algorithms and then compares these algorithms with each other to obtain an algorithm that shows the best performance. This part of the study provides brief information about the classification algorithms used herein.

3.1.1. Decision Trees

Decision tree algorithms divide a dataset into small subsets based on various features and aim to classify the data within each subset. Like branches of a tree, classes are further divided into sub-branches. The data to be classified reach the root of the tree, progress toward the sub-branches based on conditions, and become members of the closest class according to their values. The accuracy of the rules forming the classes affects the algorithm's performance.

Information gain is calculated during root selection. Information gain is utilized to determine how much information a feature provides in the classification problem. The feature with the highest information gain becomes the root. This process continues until all features are exhausted. Information gain is calculated using entropy. Entropy represents the probability of uncertainty or disorder occurring. Although decision tree algorithms have many advantages, they also have disadvantages. Decision tree algorithms can be unreliable because they have high variance, especially when the data set is too noisy. Even the slightest change in the data set can lead to large differences in tree structures. In addition, these algorithms are prone to overfitting and can capture noise in the data set as if it were true data.

3.1.2. Naïve Bayes

The Naïve Bayes algorithm is a machine learning algorithm commonly used in statistical classification problems and provides a probabilistic approach. The Naïve Bayes algorithm is quite advantageous as it does not require too much data for training (Nurdina & Puspita, 2023). Training data are calculated according to Bayes' theorem, and probability percentages are determined using the concept of probability to calculate the relationship between the features and classes in the dataset. New data are then classified based on these probability values. The assumption of the conditional independence of features in the Naïve Bayes algorithms may cause this algorithm to achieve poor performance, considering that real-life datasets may be related to each other. In addition, these algorithms can show high performance over datasets with simple distributions (e.g., Gaussian distribution) while being unable to provide optimum performance over complex data.

3.1.3. Random Forest

The random forest algorithm is a powerful machine learning method commonly used for classification or regression problems. It achieves classification by employing multiple decision trees with the goal of improving classification accuracy. Random decision trees are selected from the dataset to form a forest, thereby enhancing adaptability to the data and obtaining more accurate results over datasets. These algorithms are less prone to overfitting compared to decision tree algorithms. However, using datasets with a large number of data may result in a lot of memory consumption, because these algorithms store more than one decision tree in their memory.

3.1.4. Support Vector Machine (SVM)

SVM is a classifier used for classifying data or performing a regression process. The SVM algorithm draws a boundary to separate two different groups on a plane. This boundary is drawn as far as possible from the members of both groups and is referred to as the margin. The margin represents the gap between the classes and is adjusted in classifications based on the drawn boundaries. Test data belong to the group that is closer to the boundaries. When drawing the boundary, two lines are drawn close to each group. By bringing these drawn boundaries closer to each other, a common boundary is obtained with the aim of classify the data the most accurately. However, these algorithms may experience problems in datasets where each parameter has a different amount of data. In these cases, SVM can classify in favor of the parameter with the most data. This may cause poor performance in correctly classifying parameters with a small amount of data. In addition, SVM algorithms may consume a lot of memory when classifying data sets that contain many features and data. Therefore, it may not be a good option for classifying data sets with high amounts of data.

3.1.5. Logistic Regression

Logistic regression is a supervised learning algorithm that is used when the outcome can only take one of two values. It can be used for both classification and clustering and is employed to predict the probability of data points belonging to two different classes, typically represented as 0 and 1. Logistic regression is commonly used to address binary classification problems. Because the logistic regression algorithm assumes a linear relationship between the independent variables and the log ratios of the obtained result, it may not show high performance when classifying data with complex decision boundaries. Despite this drawback, the logistic regression algorithm is very suitable for use in cases where the outcome variable has two different categories. In addition, having low variance makes this algorithm more suitable for use regarding complex data by reducing the risk of overfitting.

The most interpretable classification algorithms are decision tree and Naïve Bayes algorithms, but they do not perform as well as other algorithms at predicting missing data. The random forest algorithm offers the most advanced prediction performance, and SVMs are more powerful than other algorithms at finding complex decision boundaries. Logistic regression may be the best option for binary classification scenarios but is not the best option for nonlinear relationships. In general, all classification algorithms have their strengths and weaknesses, and this study compares and analyzes all the mentioned classification algorithms over the study's own data set.

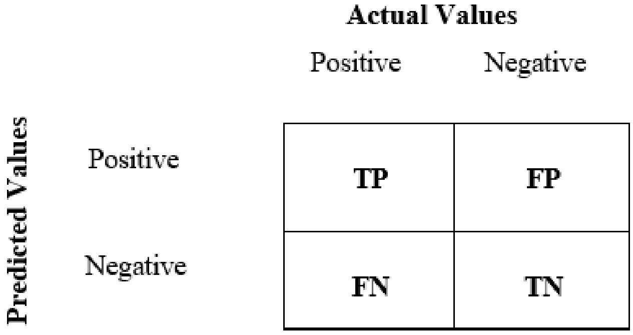
3.2. Performance Metrics

Performance metrics and the confusion matrix are crucial tools used to evaluate the performance of classification models and to analyze classification results in more detail. The confusion matrix is employed to provide a more detailed analysis of classification results by visualizing the relationship between true classes and the classes predicted by the model. These tools help assess the classification capabilities of a model and to identify the measures necessary for improving performance.

After the classification process, the study utilizes such performance metrics as the confusion matrix, accuracy, precision, recall, and F1 score to evaluate the performance of the employed methods, with the chosen performance metrics being explained as follows.

3.2.1. Confusion Matrix

The performance measurement of real and predicted data for a classification problem is represented by a confusion matrix consisting of four cells representing four different values, as illustrated in Fig. 2.



		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

Figure 2. The confusion matrix.

True Positive (TP): Positively predicted and actually positive. Correctly predicted.
False Positive (FP): Predicted as positive but actually negative. Incorrectly predicted.
False Negative (FN): Predicted as negative but actually positive. Incorrectly predicted.
True Negative (TN): Negatively predicted and actually negative. Correctly predicted.

3.2.2. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

3.2.3. Precision

Precision is found by dividing the total predicted positive rate of the classification model with the true positives. A model with a high precision value shows that the model can make predictions with high accuracy. This is particularly useful for reducing the number of false positives. The precision rate is calculated as shown in Equation 2.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3.2.4. Recall

Recall, also known as sensitivity or true positive rate, is the ratio of true positive predictions to the total number of actual positive instances. It measures how many of the actual positive instances were correctly identified. The recall rate is calculated as shown in Equation 3.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

3.2.5. F1 Score

The F1 score is calculated by taking the harmonic mean of the precision and recall metrics. This is used to achieve a balanced classification performance. The F1 score is calculated as shown in Equation 4.

$$F1 = \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

4. METHOD

This study conducts attacks on computer systems, classifies the executed attacks using machine learning methods, and subsequently compares the performance results of the employed classification methods. The study has tested the executed attacks and the algorithms on a computer running an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 2.59 GHz. Two virtual machines, one running Linux and the other Windows 7, were installed on the computer described above. The system on which the attacks were executed is a virtual machine running the Kali Linux operating system, while the target machine for the attacks is a virtual machine running the Windows 7 operating system.

The packets of the attacks on the virtual Windows 7 machine were intercepted using Wireshark and saved in .pcap format. The CICFlowMeter tool was utilized to analyze and extract features from the .pcap files obtained with Wireshark, and then the data were converted to .csv format for classification purposes. The Python programming language was employed to perform the classification using machine learning algorithms. The detailed information about the used software is provided in Table 1.

This study classifies the attacks on computer systems using machine learning algorithms and compares their performances. The flowchart for the proposed method is illustrated in Fig. 3.

The first stage carried out attacks in a virtual environment and obtained the ,pcap files using Wireshark. The CICFlowMeter tool was utilized to extract features from the obtained .pcap files which were then convert to .csv format. The resulting .csv files were merged to create a single .csv file forming the dataset. Attacks with a low number of instances in the dataset were removed, infinite values were converted to not-a-number (NaN) values, and then NaN

Table 1. *The Software Programs Used in the Study*

Software	Developer	Purpose of Use	Output Format	Source
Wireshark	Gerald Combs	Captures data packets by listening to network traffic, analyzes them, and saves them as .pcap files.	.pcap	https://www.wireshark.org/
CICFlowMeter	Canadian Institute for Cybersecurity	It has been used to analyze network traffic and extract its features.	.csv	https://github.com/ISCX/CICFlowMeter
Python	Guido van Rossum	It is an open-source programming language. It has been used for the classification process.		https://www.python.org/

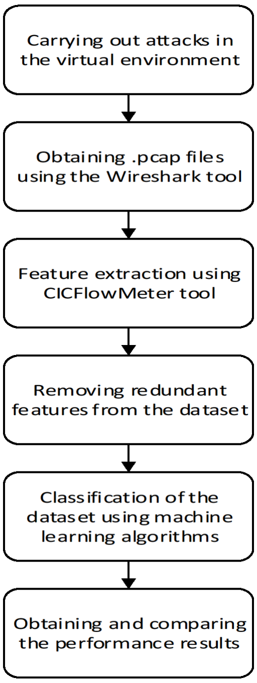

Carrying out attacks in the virtual environment
Obtaining .pcap files using the Wireshark tool
Feature extraction using CICFlowMeter tool
Removing redundant features from the dataset
Classification of the dataset using machine learning algorithms
Obtaining and comparing the performance results

Figure 3. The flowchart for the proposed method.

values were also eliminated from the dataset. This process ensured the correction of unnecessary, corrupted, or missing data. The dataset contains 80 features.

To perform the classification process on the dataset, 20% of the data was set aside for testing and 80% for training. Classification was carried out using classification algorithms. The performance results are compared using performance metrics.

4.1. Attacks Performed on the Virtual Machine

Cyberattacks were carried out to create a data set. Six different types of attacks were executed (i.e., Brute Force, DoS ICMP Flood, DoS Syn Flood, Syn Scan, UDP Scan, and Man in the Middle). Before initiating the attacks, Wireshark was run on the target machine. The Hydra tool was used for executing the Brute Force attack, and the Bettercap tool was employed for implementing the Man in the Middle attack. Hping3 was utilized for conducting the DoS attacks, and Nmap was employed for executing the Scan attacks. In addition to the attack data, normal traffic was included as a separate class in the dataset, resulting in a total of seven classes (Table 2) with numerical labels assigned accordingly.

Table 2. *Class Labels and Descriptions*

Labels	Classes	Descriptions	Software Programs Used
0	Normal	Normal network traffic has been used.	The network was eavesdropped on using Wireshark.
1	BruteForce	Session login information was obtained using the user information method over FTP, RDP, SSH protocols.	Session information was obtained using the Hydra tool.
2	Man in The Middle	Communication network was eavesdropped on and manipulated.	The Bettercap tool was used to eavesdrop on and manipulate communication on the network.
3	DoS_Syn_Flood	Intensive SYN packets were sent to the target to deplete resources.	Network traffic was manipulated using the Hping3 tool.
4	DoS_ICMP_Flood	Intensive ICMP packets were sent to the target to deplete resources.	Network traffic was manipulated using the Hping3 tool.
5	SYN_Scan	Port scanning was performed by sending SYN packets to determine the TCP connection status of the target.	The Nmap tool was used to determine the open ports of the target system.

4.2. Dataset

This stage collects the raw data by intercepting network traffic packets. Before initiating any cyberattacks, the general network traffic was monitored through Wireshark, and the recorded .pcap files were labeled as normal. Six different attacks were executed targeting specific objectives, and Wireshark was used to intercept the network attack packets. The obtained .pcap files were labeled according to the type of attack. Features were extracted from all obtained .pcap files using CICFlowmeter software, and then the .csv files were created for the dataset. The obtained .csv files were merged to create a single .csv file to form the dataset. Low-occurrence attacks were removed from the dataset, infinite values were converted to NaN values, and then the NaN values were also eliminated from the dataset. The final dataset comprises 8,094 instances, with the quantity of data for each class provided in Table 3.

Table 3. *Class-Based Data Quantities*

Class	Amount of Data
DoS_Syn_Flood	2,559
UDP_Scan	1,916
normal	1,131
BruteForce	1,118
SYN_Scan	1,002
DoS_ICMP_Flood	315
Man in The Middle	53

Initially, the dataset contained a total of 84 features; however, features were removed that could negatively impact or that were ineffective for the classification process. The removed features were generally string values. This process aimed to address unnecessary, corrupted, or missing data. In the final version of the dataset, 80 features were retained. To perform the classification over the dataset, 20% of the data were designated as the test set, and 80% as the training set. The features extracted from the dataset are based on the work of Kilincer et al. (2022).

4.3. Classification

The machine learning algorithms used for classifying the attacks against the target system include the random forest, SVM, Naïve Bayes, logistic regression, and decision tree methods. Performance metrics were employed to evaluate the models' effectiveness. Multiclass classification was used for classification and created a total of seven classes comprising six attack types and one normal class. The selection of these classification methods is based on their different capabilities that are tailored to specific features and requirements. Machine learning algorithms are widely utilized in tasks such as classification and regression and are particularly effective when dealing with high-dimensional and complex data. These methods often come with error-correction capabilities and are resistant to overfitting the data, making them suitable for applications in various domains. The study now evaluates the differences between classification models and determines which model demonstrates the best performance.

5. EXPERIMENTAL RESULTS

This study has involved creating a dataset, splitting it into training and testing sets, and applying the specified classification models to the dataset. After successful training, the study evaluated the results from the selected machine learning algorithms.

The random forest model exhibits high accuracy when predicting the normal class. It shows good performance with high accuracy at predicting attacks for the BruteForce, Man in the Middle, DoS_Syn_Flood, and SYN_Scan classes. The model demonstrates high accuracy at predicting UDP_Scan attacks. However, for DoS_ICMP_Flood attacks, the model exhibits lower performance compared to the other classes of attacks. The random forest model's overall accuracy rate has been calculated as 94.87%, indicating a high ability to make accurate classifications. The performance characteristics of the random forest classification method are presented in Table 4 and Fig. 4.

Table 4. *The Performance Metrics of the Random Forest Algorithm*

Labels	Classes	Precision	Recall	F1-Score
0	Normal	0.98	0.85	0.91
1	BruteForce	1.00	1.00	1.00
2	Man in The Middle	1.00	1.00	1.00
3	DoS_Syn_Flood	1.00	1.00	1.00
4	DoS_ICMP_Flood	0.61	0.61	0.61
5	SYN_Scan	1.00	1.00	1.00
6	UDP_Scan	0.86	0.93	0.89

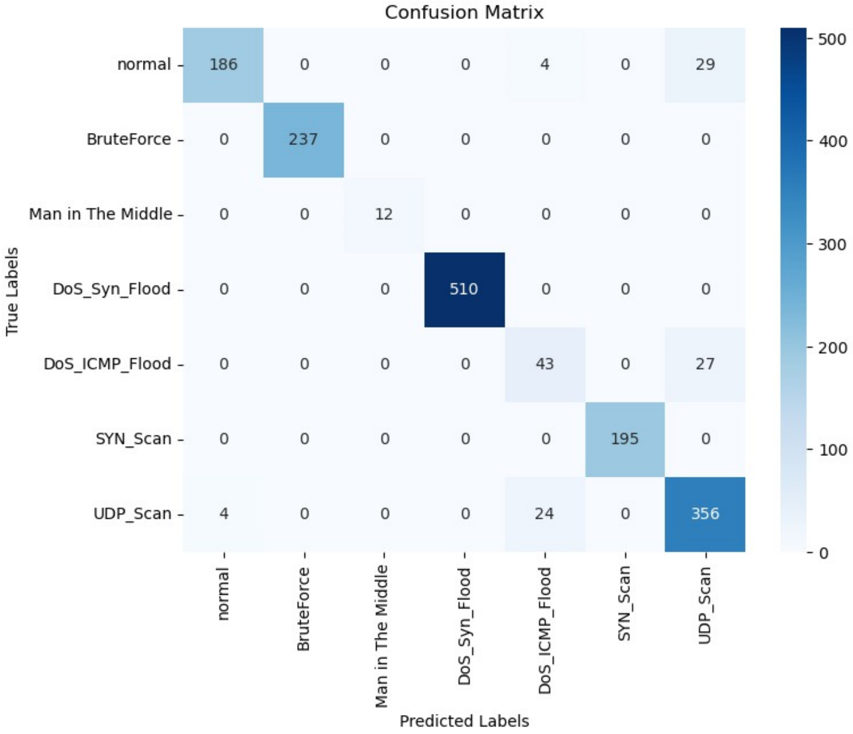


Figure 4. The confusion matrix of the random forest algorithm.

The Naïve Bayes model demonstrates a moderate accuracy when predicting the normal class. Precision, recall, and F1-score for DoS_Syn_Flood and SYN_Scan classes are 1.00, indicating the Naïve Bayes model to perform well at predicting these attacks. The model shows a moderate accuracy at predicting DoS_ICMP_Flood and UDP_Scan attacks. However, it exhibits lower performance at predicting Man in the Middle attacks compared to other classes. The overall accuracy rate of the Naïve Bayes model is calculated as 79%, indicating a moderate level of performance. The Naïve Bayes algorithm's performance values are presented in Table 5 and Fig. 5.

Table 5. *Naive Bayes Algorithm's Performance Values*

Labels	Classes	Precision	Recall	F1-Score
0	normal	0.44	0.79	0.57
1	BruteForce	0.95	0.93	0.94
2	Manin The Middle	0.14	0.91	0.24
3	DoS_Syn_Flood	1.00	1.00	1.00
4	DoS_ICMP_Flood	0.45	0.76	0.56
5	SYN_Scan	1.00	1.00	1.00
6	UDP_Scan	0.80	0.21	0.33

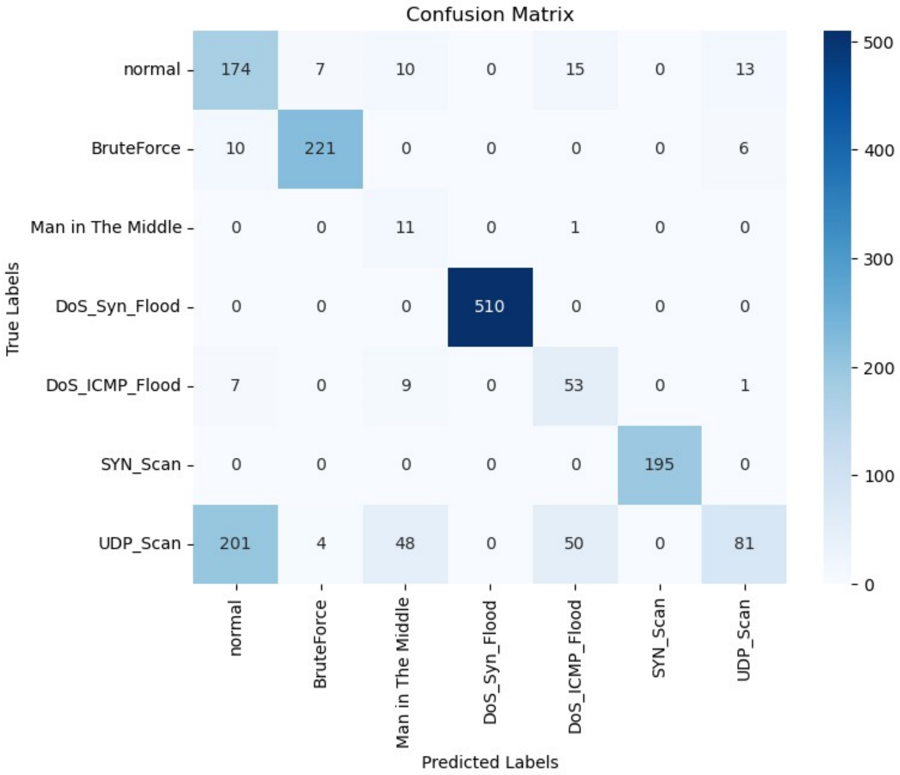


Figure 5. Naive Bayes algorithm's confusion matrix.

The logistic regression model exhibits a moderate level of performance when predicting the normal class. For Brute Force, DoS_Syn_Flood, and SYN_Scan classes, the model appears to perform well at predicting these attacks. The model shows low performance at predicting DoS_ICMP_Flood and Man in the Middle classes and demonstrates a moderate level of performance at predicting UDP_Scan attacks. The overall accuracy rate of the logistic regression model has been calculated as 88%, indicating a good level of performance. The performance values of the logistic regression classification method are presented in Table 6 and Fig. 6.

Table 6. *The Performance Values of the Logistic Regression Algorithm*

Labels	Classes	Precision	Recall	F1-Score
0	Normal	0.73	0.61	0.67
1	BruteForce	0.96	0.99	0.98
2	Man in The Middle	0.88	0.58	0.70
3	DoS_Syn_Flood	0.99	0.99	0.99
4	DoS_ICMP_Flood	0.53	0.38	0.43
5	SYN_Scan	0.99	1.00	1.00
6	UDP_Scan	0.72	0.82	0.77

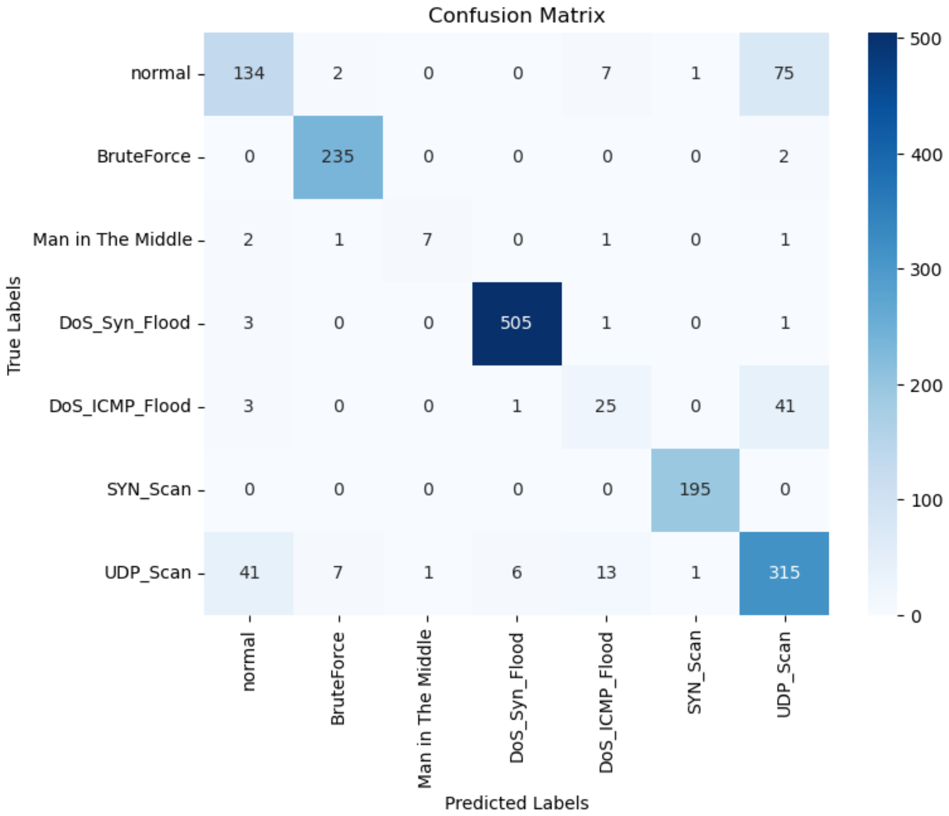


Figure 6. The confusion matrix of the logistic regression algorithm.

The SVM model exhibits a moderate level of performance when predicting the normal class. For Brute Force, DoS_Syn_Flood, and SYN_Scan classes, the SVM model seems to perform well at predicting these attacks. The model shows low performance at predicting the DoS_ICMP_Flood class and demonstrates a moderate level of performance at predicting UDP_Scan attacks. The SVM model made no correct predictions for the Man in the Middle attacks. The overall accuracy rate of the SVM model has been calculated as 85%, indicating the model to generally perform well. The performance values of the SVM method are presented in Table 7 and Fig. 7.

Table 7. *The Performance Values of the SVM Algorithm*

Labels	Classes	Precision	Recall	F1-Score
0	Normal	0.74	0.55	0.63
1	BruteForce	0.88	1.00	0.94
2	Man in The Middle	0.00	0.00	0.00
3	DoS_Syn_Flood	0.97	1.00	0.98
4	DoS_ICMP_Flood	0.74	0.37	0.50
5	SYN_Scan	1.00	1.00	1.00
6	UDP_Scan	0.70	0.81	0.75

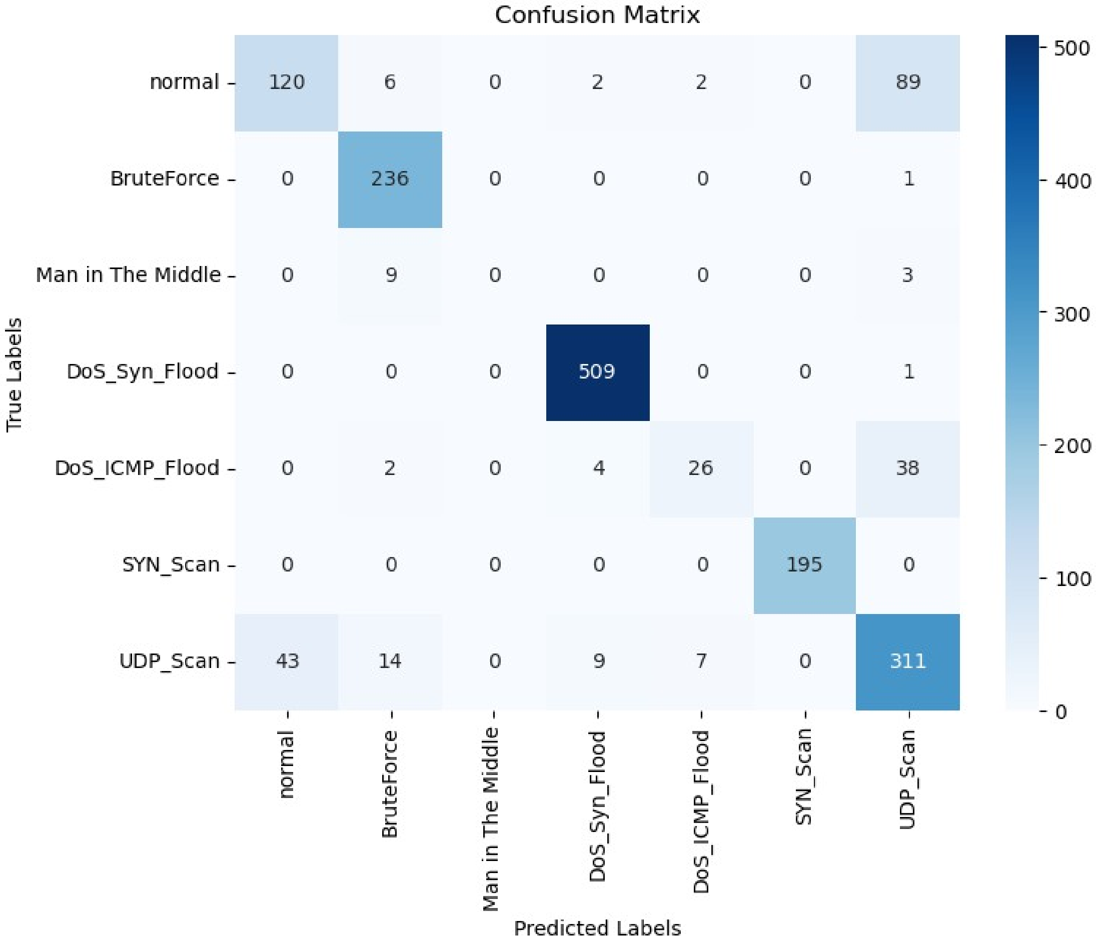


Figure 7. The confusion matrix of the SVM algorithm.

The decision tree model demonstrates high-level performance when predicting the normal class. For Brute Force, DoS_Syn_Flood, SYN_Scan, and UDP_Scan classes, the decision tree model appears to perform well at predicting these attacks. The model shows moderate-level performance at predicting the DoS_ICMP_Flood and Man in the Middle classes. The overall accuracy rate of the decision tree model has been calculated as 94%, indicating the model to generally perform well. The performance values of the decision tree method are presented in Table 8 and Fig. 8.

Table 8. *The Performance Values of the Decision Tree Algorithm*

Labels	Classes	Precision	Recall	F1-Score
0	Normal	0.84	0.88	0.86
1	BruteForce	1.00	1.00	1.00
2	Man in The Middle	1.00	0.92	0.96
3	DoS_Syn_Flood	1.00	1.00	1.00
4	DoS_ICMP_Flood	0.60	0.54	0.57
5	SYN_Scan	1.00	1.00	1.00
6	UDP_Scan	0.88	0.88	0.89

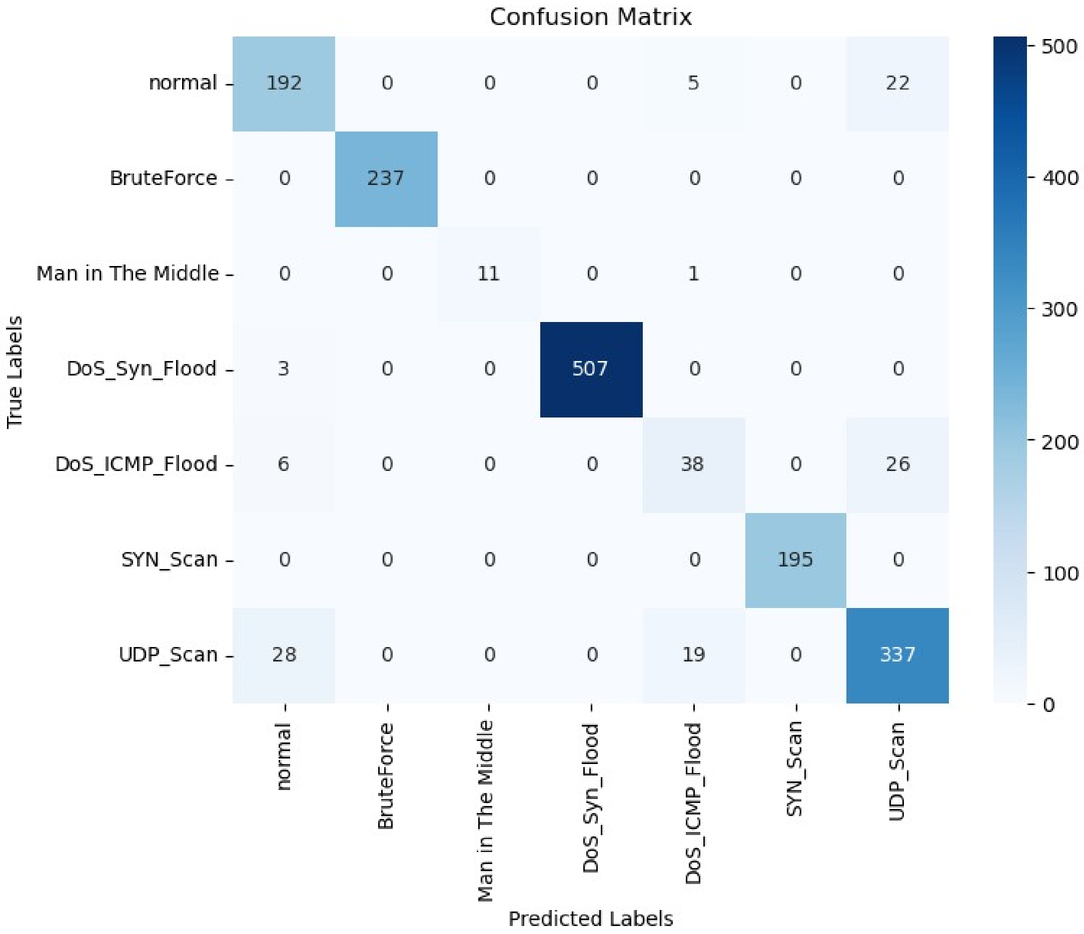


Figure 8. The confusion matrix of the decision tree algorithm.

When comparing the used classification algorithms and looking at the results in Table 9, the random forest classifier is sene to have achieved the highest accuracy rate and to also exhibit good performance in terms of precision, recall, and F1 score. The Naïve Bayes classifier appears to have the lowest performance compared to the other models.

Table 9. *Comparison of the Classifiers Used*

Classifiers	Accuracy	Precision	Recall	F1-Score
Random Forest	0.948	0.930	0.918	0.924
Naive Bayes	0.789	0.853	0.789	0.789
Logistic Regression	0.881	0.877	0.881	0.876
SVM	0.856	0.851	0.856	0.846
Decision Tree	0.941	0.941	0.941	0.941

6. CONCLUSIONS AND FUTURE DIRECTIONS

The field of network forensics addresses issues such as detecting cybercrimes, analyzing attacks, and ensuring network security. The utilization of machine learning techniques in this domain demonstrates significant advantages for obtaining, analyzing, detecting, classifying, and implementing security measures and can lead to the development of robust products. This study discusses threats and attacks on computer systems, with some of these attacks being simulated in a virtual environment. The study executed various types of cyberattacks on a simulated target computer in a virtual environment. Feature extraction from network packets was performed to prepare a dataset for classification. Different machine learning classification algorithms were applied to the created dataset to compare their abilities at accurately classifying attacks.

While some algorithms were successful at classifying certain attacks, others did not achieve the same level of success. Generally, lower performance was observed at classifying DoS_ICMP_Flood and Man in the Middle attacks. The imbalance in dataset numbers is considered a potential cause for the lower performance in these attacks. Collecting more data for such attacks or employing different feature extraction methods could enhance the classification performance. When evaluating the performances of the classification algorithms, the random forest classifier stands out for having achieved the highest accuracy rate and demonstrating good precision, recall, and F1 score.

To assess efficiency, measuring the training and testing times of the selected machine learning algorithms would be beneficial. This approach allows for the evaluation of critical factors such as accuracy and processing speed, thus facilitating an objective comparison for determining the most suitable method.

This study aims to contribute to detecting attacks on computer systems and to developing security strategies. The obtained results can help identify which algorithms perform best on a dataset and which algorithms more accurately detect specific cyberattacks. As a result, the study provides guidance to network forensic experts and security professionals on how to effectively detect, analyze, and take preventive measures against attacks. This study may contribute to enhancing security measures in the field of network forensics and the formulation of strategies to combat cybercrimes.

In order to secure computer systems, stay ahead of industry innovations and tailor security policies is imperative. Contemporary machine learning techniques that have been widely adopted in the realms of cybersecurity and digital forensics play a pivotal role in detecting anomalies and are crucial for data detection and analysis.

High-performance classification algorithms such as random forest are preferable for analyses in digital forensics within network security. To enhance the performance of machine learning algorithms with suboptimal efficiency, further research and improvement efforts are essential. Some attacks exhibited low performance regarding classification, suggesting that collecting more data or employing different feature extraction methods could enhance the applications used for feature extraction.

Future studies are recommended to involve datasets with increased data volume and to conduct analyses for various operating systems. Alongside the continuous development of security technologies and the reinforcement of security measures, this study proposes holding awareness campaigns and training programs on cybersecurity to augment awareness among individuals.

This study's approach has contributed to the perpetual advancement of security technologies and the overall reinforcement of cybersecurity measures.

Peer Review: Externally peer-reviewed.

Author Contributions: Conception/Design of Study- F.Y., B.G.; Data Acquisition- F.Y., F.E.; Data Analysis/Interpretation- F.Y., B.G.; Drafting Manuscript- B.G., F.Y.; Critical Revision of Manuscript- F.E.; Final Approval and Accountability- F.Y., B.G., F.E.; Supervision- F.E.

Conflict of Interest: The authors have no conflict of interest to declare.

Grant Support: The authors declared that this study has received no financial support.

ORCID IDs of the author

Firdevs Yıldız	0000-0002-6101-9798
Batuhan Gül	0009-0007-1772-5373
Fatih Ertam	0000-0002-9736-8068

REFERENCES

Aamir, M., Rizvi, S. S. H., Hashmani, M. A., Zubair, M., & Usman, J. A. . (2021). Machine Learning Classification of Port Scanning and DDoS Attacks: A Comparative Analysis. *Mehran University Research Journal of Engineering and Technology*. https://doi.org/10.22581/muet1982.2101.19

Ahmetoğlu, H., & Daş, R. (2021). Makine Öğrenmesi Yöntemleri Kullanarak Web Uygulama Saldırılarının Tespitinde Genetik Öznitelik Seçimi Yaklaşımı. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi.* https://doi.org/10.54525/tbbmd.1018465

Akbal, E., Doğan, Ş., Tuncer, T., & Atalay, N. S. (2019). Adli Bilişim Alanında Ağ Analizi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. https://doi.org/10.17798/bitlisfen.479303

AlZubi, A. A., Al-Maitah, M., & Alarifi, A. (2021). Cyber-attack detection in healthcare using cyber-physical system and machine learning techniques. *Soft Computing*. https://doi.org/10.1007/s00500-021-05926-8

Ashton, J. J., Young, A., Johnson, M. J., & Beattie, R. M. (2023). Using machine learning to impact on long-term clinical care: principles, challenges, and practicalities. *Pediatric Research*. https://doi.org/10.1038/s41390-022-02194-6

Aslan, O., & Yilmaz, A. A. (2021). A New Malware Classification Framework Based on Deep Learning Algorithms. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3089586

Başlar, Y. (2020). Adli Bilişim Sürecinde Karşılaşılan Sorunlar ve Çözüm Önerileri. *Türkiye Barolar Birliği Dergisi, 32*(148), 47–76. Retrieved from https://app.trdizin.gov.tr/makale/TXpZeU5EUXpNdz09/adli-bilisim-surecinde-karsilasilan-sorunlar-ve-cozum-onerileri

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*. https://doi.org/10.1093/aje/kwz189

Dina, A. S., & Manivannan, D. (2021). Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things (Netherlands)*, 16(August). https://doi.org/10.1016/j.iot.2021.100462

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications. https://doi.org/10.1016/j.jisa.2019.102419

Hunt, R. (2012). New developments in network forensics-Tools and techniques. *IEEE International Conference on Networks*, ICON. https://doi.org/10.1109/ICON.2012.6506587

İnce, C., İnce, K., Hanbay, D., Üniversitesi, İ., İşlem, B., Başkanlığı, D., . . . Bölümü, M. (2021). Saldırı Tespit Sistemlerinde Sınıflandırma Yöntemlerinin Kıyaslanması. *Dergipark.Org.Tr*, (1), 1–11. Retrieved from https://dergipark.org.tr/en/pub/bbd/issue/59753/791939

Karaman, M. S., Turan, M., & Aydin, M. A. (2021). Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması. *European Journal of Science and Technology.* https://doi.org/10.31590/ejosat.1115825

Kilincer, I. F., Ertam, F., & Sengur, A. (2022). A comprehensive intrusion detection framework using boosting algorithms. *Computers and Electrical Engineering.* https://doi.org/10.1016/j.compeleceng.2022.107869

Krishna Suryadevara, C. (2023). Issue 4 Diabetes Risk Assessment Using Machine Learning: A Comparative Study of Classification Algorithms. *International Engineering Journal For Research & Development, 8*(4). Retrieved from www.iejrd.com

Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports.* https://doi.org/10.1016/j.egyr.2021.08.126

Nancy Noella, R. S., & Priyadarshini, J. (2023). Machine learning algorithms for the diagnosis of Alzheimer and Parkinson disease. Journal of Medical Engineering and Technology. https://doi.org/10.1080/03091902.2022.2097326

Nurdina, A., & Puspita, A. B. I. (2023). Naive Bayes and KNN for Airline Passenger Satisfaction Classification: Comparative Analysis. *Journal of Information System Exploration and Research.* https://doi.org/10.52465/joiser.v1i2.167

Özekes, S., & Karakoç, E. N. (2019). Makine Öğrenmesi Yöntemleriyle Anormal Ağ Trafiğinin Tespit Edilmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*. https://doi.org/10.29130/dubited.498358

Pallathadka, H., Wenda, A., Ramirez-Asís, E., Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. *Materials Today*: *Proceedings*. https://doi.org/10.1016/j.matpr.2021.07.382

Qureshi, S., Tunio, S., Akhtar, F., Wajahat, A., Nazir, A., & Ullah, F. (2021). Network Forensics: A Comprehensive Review of Tools and Techniques. *International Journal of Advanced Computer Science and Applications.* https://doi.org/10.14569/IJACSA.2021.01205103

Radivilova, T., Kirichenko, L., Ageiev, D., & Bulakh, V. (2019). Classification methods of machine learning to detect DDoS attacks. *Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019.* https://doi.org/10.1109/IDAACS.2019.8924406

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. *Future Generation Computer Systems.* https://doi.org/10.1016/j.future.2020.02.017

Sharma, N., Sharma, R., & Jindal, N. (2021). Machine Learning and Deep Learning Applications-A Vision. *Global Transitions Proceedings.* https://doi.org/10.1016/j.gltp.2021.01.004

Shaukat, K., Luo, S., Chen, S., & Liu, D. (2020). Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective. *1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings*. https://doi.org/10.1109/ICCWS48432.2020.9292388

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. *IEEE Access.* https://doi.org/10.1109/ACCESS.2020.3041951

Wani, A. R., Rana, Q. P., Saxena, U., & Pandey, N. (2019). Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques. *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019.*

https://doi.org/10.1109/AICAI.2019.8701238

Zhang, X., Chen, J., Zhou, Y., Han, L., & Lin, J. (2019). A Multiple-Layer Representation Learning Model for Network-Based Attack Detection. *IEEE Access.* https://doi.org/10.1109/ACCESS.2019.2927465

How cite this article

Yıldız, F., Gül, B., & Ertam, F. (2024). Network forensics analysis of cyber attacks on computer systems using machine learning techniques. *Acta Infologica, 8*(1), 34-50. https://doi.org/10.26650/acin.1444470