SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 7, No. 2, 2024, 203-216
DOI: 10.35377/saucis...1462721
RESEARCH ARTICLE

Network Forensics Analysis of Cyber Attacks Carried Out Over Wireless Networks Using Machine Learning Methods

İmran Kaçan[1], Batuhan Gül[1], Fatih Ertam[1]

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

Corresponding author:
Batuhan Gül, Fırat University, Faculty of Technology, Digital Forensics Engineering, Elazığ, Türkiye
b.gul@firat.edu.tr

Article History:
Received: 01.04.2024
Accepted: 04.06.2024
Published Online: 23.08.2024

ABSTRACT
As technology advances, the frequency of attacks targeting technological devices has surged. This rise in cyber threats poses a constant risk to the devices we rely on. Any device connected to a network becomes vulnerable to exploitation by attackers. Given the extensive interconnectedness of devices in network environments, this research endeavors to address this pressing issue. The aim of this study is to analyze and classify network traffic generated during potential cyber attacks using various classification algorithms. By subjecting a simulated environment to different cyber attack scenarios, we extract the distinctive features of network packets generated during these attacks. Subsequently, we employ widely used classification algorithms to train and analyze the obtained data. For the comparison of models, more than 7000 attack data instances were employed. At the conclusion of the comparison, the Gradient Boosting algorithm achieved the highest accuracy value, reaching 91%, whereas the Naive Bayes algorithm obtained the lowest accuracy, reaching 74%.

Keywords: Network forensics, Cyber security, Machine learning

1. Introduction

Communication between at least two computers is called computer networks [1]. In the interconnected realm of the cyber world, computer networks can be inherently vulnerable to various types of attacks. The attack examples are such as man-in-the-middle attacks, denial-of-service attacks, distributed denial-of-service attacks, and malicious software injection [2].

The significant increase in internet usage has accompanied the advancement of technology. Numerous devices in our surroundings are now connected to networks. Furthermore, the rise in technology has led to an increase in digital crimes.

The widespread adoption of the evolving technological infrastructure implies that these systems are exposed to significant risks. While the increase in technological structures facilitates human life, it also allows threat elements access to various sources and the potential exploitation of system vulnerabilities [3]. Consequently, the emergence of individuals intending to inflict harm on these systems, coinciding with the utilization of technological devices and network technologies, has given rise to a category of crimes known as cybercrimes.

In the realm of cybercrimes, data recorded in electronic/magnetic fields is referred to as digital evidence. Various types of digital evidence exist, such as photos, videos, server log files, web history, data files and registration logs. The majority of these data are transmitted over the network. One crucial form of analysis is network analysis.

The general purpose of the programs/tools used for network analysis is to listen to network traffic, capturing incoming and outgoing packets during the listening process for network analysis. A network consists of two or more devices such as computers, servers, and network devices, sharing resources like printers, engaging in file exchange, or permitting electronic communication. Additionally, it can be asserted that the most effective network security method involves managing access to the network [4]. In implementing these security measures, a thorough understanding of the user profiles connected to the network is imperative. Having command over user profiles facilitates the work of network administrators during the authorization processes. When authorization is tailored to the user profile, managing the network becomes more straightforward. It is imperative that each user does not have unrestricted access to every network. Access to network resources should be granted only to authorized users, preventing malicious activities by restricting unauthorized access to the network. Unused ports should be closed, and structures should be left open based on user needs. Authentication methods

must be activated. Additionally, understanding various attack types is crucial for defining security policies. Considering all these aspects, the significance of forensic network analysis becomes evident. Security measures in this field should be enhanced, and the number of educated individuals in the domain should be increased.

We propose obtaining the necessary data for network forensic analysis through the application of machine learning models. Previous studies have addressed analysis topics in wireless networks using machine learning.

The utilization of machine learning models has been proposed by Dhanya et al. [5] for the detection of cyber attacks targeting wireless networks. In the suggested methodology, a comparison of nine distinct machine learning algorithms has been conducted, and this comparative analysis has been executed on the UNSW-NB15 dataset. Among the compared machine learning models, the Decision Tree algorithm exhibited the highest performance, achieving a remarkable accuracy of 99.05% and a recall value of 99%. Conversely, the SVM algorithm demonstrated the least favorable performance, attaining an accuracy of 95.17% and a recall value of 93%. The authors suggested converting network data into images to improve the detection performance of the models.

Ahmad et al. [6] discussed machine learning methods in wireless sensor networks and the potential for detecting and classifying attacks on wireless networks using these methods. The authors have asserted the effectiveness of employing machine learning models in wireless sensor networks. Furthermore, they have postulated that the utilization of Software-Defined Networking (SDN) technology will enhance the performance of the machine learning model.

Mughaid et al. [7] proposed using machine learning models to detect attacks in 5G wireless networks. The authors developed a simulation software and obtained attack data with this simulation software. The OMNeT++ software has been utilized for simulation purposes, wherein a Dropping attack scenario was created to compare the performance of various machine learning models. In the conducted experiments, the Logistic Regression model achieved the highest accuracy rate at 95.7%, while the Naive Bayes model reached the lowest accuracy rate at 76.7%.

A survey study has been conducted by Waqas et al. [8] utilizing artificial intelligence and machine learning to classify cyber threats in wireless networks. In the article, the authors initially enumerate the cyber threats that necessitate attention, subsequently comparing and categorizing these threats for analysis. The second section of the article discusses potential defense mechanisms against the identified threats, utilizing machine learning methods and artificial intelligence models.

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

- Cyber attacks are being conducted on computer systems through wireless networks, and following the execution of these attacks, network traffic is recorded using Wireshark to construct our dataset.
- We analyze the obtained network packets and, through feature extraction, employ machine learning approaches to classify these network packets.
- In classifying data within our created dataset, the most successful model is the Gradient Boosting model, achieving an accuracy rate of 91%. Conversely, the model with the lowest accuracy rate is the Naive Bayes model.

In the first part of our study, we provide general information about cyber attacks on wireless networks and the damage these attacks can cause, and we compile previous studies on the use of machine learning techniques in wireless networks. In the second part, we detail the cyber attacks that can be made on wireless networks. In the third part of our study, we provide detailed information about machine learning classifiers and introduce our method. In the fourth section, we classify and compare results obtained from machine learning techniques. In the last part of our study, we include the conclusions and suggestions.

2. Cyber Threats on Wireless Networks

With the increasing importance of portability, the utilization of wireless networks has also seen a surge. Alongside portability, wireless access points are employed to expand networks [9] .Numerous users connect to these expanded networks. Wireless networks, providing convenient usage for a large number of users, are susceptible to cyber attacks. For these structures to be considered secure, they must fulfill conditions of authenticity, confidentiality, integrity, and usability [10].

Cyber attacks occur across different layers of the OSI model. The first layer of the OSI model, known as the Physical layer, facilitates the transfer of data packets between devices. This transfer is achieved through electrical or light signals. The other layers operate in a manner dependent on the Physical layer.

The next layer, the Data Link layer, separates bits from the Physical layer into packets when transferring from one device to another. This layer controls the physical addresses of devices to ensure the accurate delivery of data packets. The data transmission process occurs bit by bit.

The Network layer, which utilizes the IP protocol for communication, performs addressing and routing operations during the transmission of data.

In the Transport layer, data is segmented and reassembled at the destination. The Transport layer header is added to ensure the correct delivery of data.

The Session layer manages the opening, maintaining, and termination of necessary sessions between applications. To facilitate seamless data exchange, this layer ensures that the session remains open for an adequate duration during data transmission. Systems can initiate bidirectional communication when establishing a connection in this layer. Once the data transfer process is completed, the session is terminated to prevent unnecessary resource consumption.

The Presentation layer determines which protocols to use during the exchange of data packets and performs data transformation. Data is transformed into suitable formats for transfer to the Application layer.

The Application layer is the topmost layer visible to end-users. Users can provide data input in this layer, offering the clearest view for users. This layer includes various protocols that allow communication with applications such as email, instant messaging, and file transfer.

Table 1 demonstrates cyber attacks against OSI layers.

Table 1. Cyber Attacks Against OSI Layers

Layer	**Attack Type**
Physical Layer	Eavesdropping, Jamming, Side-Channel Attacks, Random Interference, Timing Attack
Data Link Layer	MAC Spoofing, Identify Theft, man in the Middle, network Injection, Mac Flooding
Network Layer	IP Spoofing, IP Hijacking, Smurf Attack, Sinkhole Attack
Transport Layer	TCP Flooding, UDP Flooding, TCP Guessing Attack
Application Layer	Malware Attack, SQL Injection, Cross-Site Scripting, FTP Bounce

In preparation for this study, a test environment was set up to obtain the necessary data, and six wireless network attacks were conducted. Deauthentication Attack, DoS, UDP Flood, ICMP Flood, SYN Flood and Man in The Middle attacks were performed using this test environment.

2.1. Deauthentication Attack

In deauthentication attacks, which fall under the category of second-layer attacks [11], the aim is to disconnect devices connected to a wireless network by sending numerous packets. This type of attack can be considered within the class of Service Denial of Service (DoS) attacks.

The structure of a deauthentication attack, which is designed to disrupt the access of devices connected to the network, is illustrated in Figure 1.

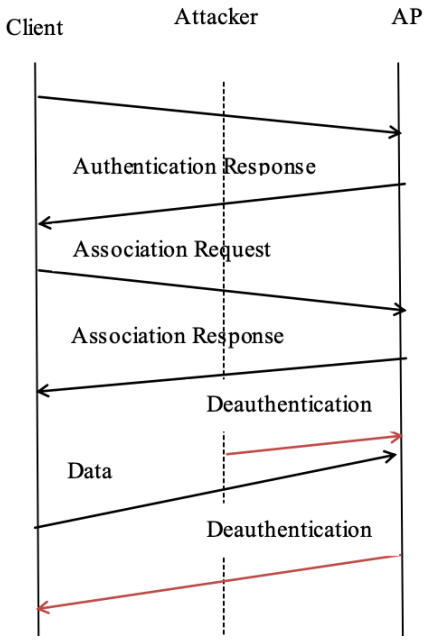

Figure 1. Deauthentication Attack [12]

Here's an overview of how a Deauthentication Attack works;

Frame Capture: The attacker puts a Wi-Fi network card into monitor mode to capture network traffic. This mode allows the attacker to see all the traffic on the network.

Sending Fake Frames: The attacker sends fake deauthentication frames to the target client or access point. These frames spoof the identity of the client or the access point. The frames cause the target client or access point to disconnect.

Disconnecting the Client: When the target client receives the fake deauthentication frame, it disconnects from the current connection. This often causes the client to try to reconnect, but if the attacker continuously sends deauthentication frames, the client keeps getting disconnected.

Service Disruption: By continuously sending deauthentication frames, the target client is repeatedly disconnected, resulting in a denial of service. This prevents the client from accessing the internet or the network.

To protect the system against such attacks, more secure network protocols such as WPA3 should be used and it is important to use IDS/IPS to monitor network traffic and block unusual packets.

2.2. DoS (Denial of Service) Attack

With the development of Internet technology, DoS attacks have become the most widely used cyber attacks [13][14].This type of attack aims to disrupt access to information/services on target devices/systems by the threat actor. The target can be any device, as well as network connections or site access. As a result of the attack, the target device/server becomes unavailable. To elaborate on the process, the threatening machine sends many requests to the target, and the target machine/server responds to these requests. Due to the overwhelming number of requests, the resources of the target device/server are depleted after a while, rendering the system unusable during this period. A decrease in system performance, temporary unavailability of web pages after a DDoS attack, and an increase in spam emails are symptoms of a DDoS attack [15].

DoS attacks are categorized as Flooding Attacks, Application Layer Attacks, Amplification attacks, and Resource Exhaustion attacks. Firewalls, IDS and IPS should be used more to protect the system from such attacks. In addition, Rate Limiting (limiting the packets coming from a source) should be done. The consequences of a DoS attack can also be mitigated by distributing traffic across multiple servers.

2.3. UDP (User Datagram Protocol) Flood Attack

Since it has become clear that UDP attacks are much faster and more effective than TCP attacks, attackers have started to perform more UDP flood attacks. The structure of the UDP protocol does not require the processing of a packet after it is received. For this reason, the attacker can disable the system by sending a very large number of packets to the target system. If the packets reached the destination, the attack was successful. In this type of attack, a randomly generated source IP is primarily created, and UDP packets are sent to random targets between main hosts. The targeted machine undergoing the attack;

- Checks whether there is an application listening on the relevant port,
- Responds with an ICMP packet stating "Destination Unreachable" when it is observed that no application is listening on the port.

When many UDP packets are sent, the target system is forced to respond with a significant number of ICMP packets. This situation can lead to the depletion of system resources, making it difficult for other clients to access the system.

To prevent such attacks, we recommend increasing the use of Rate Limiting (blocking multiple UDP packets from the same source) and IDS/IPS.

2.4. ICMP (Internet Control Message Protocol) Flood Attack

ICMP Flood Attack utilizes the Internet Control Message Protocol (ICMP) to send an echo packet to the target device, checking whether the target user is alive or not. In this type of attack, large volumes of ping packets are sent to the target device. These packets solicit a response from the target, consequently exhausting the bandwidth of the target network. During an ICMP Flood attack, the source IP can be spoofed. When the threat actor engages in IP spoofing to conceal their true identity, tracing the attack's origin becomes more challenging [16].

Effects of an ICMP Flood Attack; Network bandwidth saturation, System resource exhaustion, Service Disruption. Nowadays, these attacks are not very easy to perform because most network routers now reject packets sent to broadcast addresses on their networks.

2.5. SYN Flood Attack

During data exchanges between servers and targets on systems, the three-way handshake is witnessed. In this situation, known as the three-way handshake, the target receives the SYN packet, and information such as IP address and source connection is verified in the corresponding source table. Once these processes are completed, SYN-ACK packets are sent back to the client with pre-established identification information. In the final step, when the target receives the ACK packet, the table is queried again to verify whether the correct identification information has been received from the client by checking the acknowledgment number. If all steps are completed, the authentication is successful [17].

In SYN Flood attacks, interference occurs during this three-way handshake, initiating the attack. The goal of this attack is to overwhelm the system by sending more data than it can handle, similar to denial-of-service attacks, and prevent the establishment of connections.

In such attacks, the system creates half-open connections for SYN packets sent by the attacker and waits for a while. For this reason, network performance drops significantly and resources such as CPU and RAM used by the system are rapidly depleted.

2.6. Man in the Middle Attack

The Man-in-the-Middle (MitM) attack type aims to eavesdrop on the data between two connections. In this type of attack, not only can data be intercepted, but modifications to the data are also possible. The logic behind executing the attack can be expressed as follows: in environments with wireless network broadcasts, the network traffic is redirected through the threatening machine to eavesdrop on the data of individuals connected to this network, leading to the capture of the data. This interception occurs between the target and the network elements. These network elements can be a modem, router, server, or switch.

MitM attacks are challenging to detect as they acquire the location of clients without severing their connection to the network [18]. Attacks such as IP spoofing, DNS spoofing, HTTPS spoofing, Wi-Fi eavesdropping, and Session hijacking are sub-branches of the Man in the Middle attack. To prevent such attacks, it is important to use relatively more reliable HTTPS protocols, use multi-factor authentication (MFA), and use IDS and IPS, which are systems that detect and block abnormal traffic on the network.

3. Material and Method

During the establishment of the working environment, it was observed that there are numerous paid and free software options. Among these, the following free software have been selected for use in this study. The devices used for the test environment were selected based on the needs of the applications.

The devices used for the test environment and their details are as follows:

- Windows 10 Pro – x64 processor– 8,00 GB RAM – Computer
- Windows 10 Home – x64 processor– 4,00 GB RAM – Computer
- Tp-link – Archer C5v – AC1200 Wireless Dual Band Gigabit VoIP Router
- USB 2.0 Wireless 802.INN

The software used for the test environment is as follows;

- Oracle VM VirtualBox 6.1.16
- Debian – x64 processor – 2,00 GB RAM – Virtual Machine
- Wireshark 4.0.5
- CicFlowMeter

Windows 10 Pro was installed on the host machine, and it was transformed into a machine where cyber attacks would be executed by setting up a virtual machine. Oracle VM VirtualBox software was chosen for virtual machine installation. A USB 2.0 Wireless adapter connected the virtual machine to the wireless network.

For the execution of applications, a Windows 10 Home device was used as the target machine. The Wireshark tool, used for listening to network packets, was installed on this machine, and the network listening processes were carried out from this target machine.

The CicFlowMeter tool was utilized for feature extraction from the packets obtained with Wireshark. Details about the implementation are given in the proposed method section.

3.1. Machine Learning Applications in Cybersecurity

Machine learning, briefly defined, involves the parsing of data through specific algorithms, leading to the learning of parsed data and resulting in making judgments or predictions about a particular subject [19]. Machine learning enables us to comprehend data, and with technological advancements, it has become essential for handling vast amounts of generated data. Considering the significant increase in cyber threats in today's landscape, it is evident that the volume of data generated in this field has also reached substantial proportions.

In this study, data obtained in cybersecurity has been analyzed using machine learning methods.

3.1.1. Machine Learning Techniques

Various machine learning techniques are employed to facilitate the learning of machines in our surroundings. Since not every machine learning approach yields optimal results for all types of data, diverse machine learning techniques exist. In the data processing phase, performance metrics come into play to determine the most suitable learning technique for the data

When categorizing machine learning techniques, they can be grouped under four main headings, with numerous classification algorithms available for training. This study, however, focuses solely on the details of classification algorithms used during the application.

Supervised Learning: In supervised learning, the goal is to make inferences from labeled training data. The training data in supervised learning includes the input data and their corresponding labels. In other words, the outputs obtained during the testing phase are generated based on the information acquired from the provided datasets during training.

Unsupervised Learning: In these models, as testing processes are performed on data, the model's decision-making ability improves. This learning model attempts to learn the relationships between data based on the input data provided. Increasing the number of data and testing processes will enhance the success rate.

Semi-Supervised Learning: In this learning model, the number of labeled data is limited. Labeled data is used to predict unlabeled data. Multiple trials are conducted in this learning model, and learning is achieved from the acquired training experiences to obtain the best performance.

Reinforcement Learning: In reinforcement learning models, specific rules are employed to achieve the best results. Multiple different methods are used together in this model, and the model is created by determining the operation that yields the best results.

3.1.2. Classification Algorithms

Classification algorithms are a supervised learning technique used to determine the category of new observations using training data. These algorithms perform the learning process from the data set and classify these learnings into several classes/groups. These classifications can be referred to as labels/categories. In classification, the fundamental aim is to specify the class into which a new data point will fall. The concept of a classifier refers to an algorithm that maps given data inputs to a specific category. At the same time, the classification model can be defined as structures that evaluate input data given for training to derive certain results. A feature can be defined as an individual measurable property of an observed phenomenon.

- **Naive Bayes**

The Naive Bayes theorem, based on the review of probabilities, was first used by Thomas Bayes, and Naive Bayes Classifiers were developed using this theorem. This theorem allows the probability of the occurrence of a second event to be determined

when a certain event has occurred. In this scenario, the first event forms the evidence data, while the second event is the hypothesis.

There are three different models for this classifier: Gaussian, multinominal and Bernoulli. It is useful in cases where there is a large number of variable data. Additionally, in this classification algorithm, unlike other classification algorithms, as the number of feature data increases, the results obtained improve.

Advantages of the Naive Bayes Classifier include its ease of understanding and creation. Even when using the Naive Bayes Classifier in examples with large datasets, it can quickly complete the data training. Structurally, it is a very simple and fast algorithm [20].

The Naive Bayes Classifier has proven to deliver excellent results in various fields such as human motion recognition projects, traffic congestion projects, and medical research projects.

- **Gradient Boosting**

The Gradient Boosting algorithm can be used for both regression and classification models.

The steps of the Gradient Boosting algorithm are as follows:

Step 1. In solving a regression problem, the initial predictions for each data point are taken as the average of their values. The logarithm of the probabilities is taken, and this value is used as the probability for the class prediction.

Step 2. The loss value in predictions is calculated.

Step 3. A new decision tree is created using the predicted values. This tree is trained on the original dataset to learn.

Step 4. This new model is added to the ensemble. When making the next prediction with this value, it is implied that the first predictive value will be used along with the new decision tree.

Step 5. Steps 2 through 4 are repeated until the defined boundary for decision trees is reached or until improvement ceases after adding a new decision tree.

- **Support Vector Machines (SVM)**

Support Vector Machines (SVM) is a classification model that performs regression analysis. The supervised learning SVM model is based on statistical learning theory.

The explanatory variables are mapped into a high-dimensional space through non-linear structures, and then, an optimal hyperplane is created that effectively separates both classes. This hyperplane aims to maximize margins or the sum of the distances from each class's nearest training examples while minimizing classification errors [21].

Kernel functions are used in this classification method to transform the input data. This transformation is a process of converting input data into a high-dimensional space between two classes. The higher the separation between these data groups, the better the performance of the support vector machines.

- **K-Nearest Neighbors**

Conceptually, K-Nearest Neighbors is one of the easiest-to-understand classification algorithms. In the K-Nearest Neighbors classification algorithm, the feature values of sample data are plotted in an n-dimensional space, where n is the number of data features. Each point in the n-dimensional space is labeled with a class value. To explore the classification of an unlabeled data point, it is plotted in the n-dimensional space, and the class labels of the k nearest data points are noted. Typically, k is an odd number. The class that occurs the most among the k nearest data points is assigned as the class of the new data point. In other words, the decision is made by the vote of the nearest neighbors. One advantage of the K-Nearest Neighbors classification algorithm is its suitability for parallel processing [22].

3.1.3. Performance Metrics

In situations where the performance of a classification model needs to be evaluated for each class, class-based performance metrics can be used, aside from accuracy. Four conditions arise in binary guessing [20]:

True Positive (TP): Defined as examples that are actually positive and are correctly predicted as positive by the classifier.

False Positive (FP): Defined as examples that are actually negative but are incorrectly predicted as positive by the classifier.

False Negative (FN): Defined as examples that are actually positive but are incorrectly predicted as negative by the classifier.

True Negative (TN): Defined as examples that are actually negative and are correctly predicted as negative by the classifier.

Based on the information above, the precision value is calculated using the first equation, and the calculation of the recall value is provided in the second equation. To elaborate further: Precision is determined by dividing the total number of elements correctly predicted as positive (TP) by the sum of true and false positives (FP). In other words, it is the fraction of

correctly predicted positive instances out of all instances predicted as positive. Equation 1 and Equation 2 demonstrate the precision and recall values.

$$\boldsymbol{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \quad \textbf{(1)}$$

$$\boldsymbol{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \quad \textbf{(2)}$$

Precision indicates how many of the predicted positive instances are positive. Recall, on the other hand, is obtained by dividing the true positive instances by the total number of instances classified as positive. Specifically, false negatives are instances that the model has labeled as negative but are actually positive. Recall measures the predictive accuracy of the model for the positive class; intuitively, it assesses the model's ability to find all positive instances in the dataset.

Accuracy is one of the most popular metrics in multiclass classification. The calculation for accuracy is given in the third equation.

$$\boldsymbol{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+FP+TN+FN}} \quad \textbf{(3)}$$

Accuracy provides a general measure of how many correct predictions the model makes across the entire dataset.

The F1 Score evaluates the performance of a classification model starting from the confusion matrix. It combines Precision and Recall measurements under the concept of the harmonic mean, as seen in the equation number 4 below.

$$\boldsymbol{F1\ Score} = \frac{\mathbf{2*Precision*Recall}}{\mathbf{Precision+Recall}} \quad \textbf{(4)}$$

The formula for the F1 Score can be interpreted as a weighted average between precision and recall. The highest possible value for the F1 Score is 1, while the lowest is 0 [23].

3.2. The Proposed Method

The planned system for our study involves initiating cyber threats on wireless networks as the first step. The subsequent step involves recording the network traffic generated during the execution of cyber threats. Analyzing these recorded network packets constitutes the third step of the created methodology. In the subsequent steps, after extracting features and optimizing the process, machine learning approaches are employed for classification.

When performing the deauthentication Attack, the Wireshark tool was initially executed on the target machine to record network traffic during the attack. The network monitor mode was activated with the "airmon-ng start wlan0" command. The next step involved scanning nearby networks using the "airodump-ng wlan0" command. Following this, the attack was initiated by executing the commands "airodump-ng --channel channel number --bssid router mac address wlan0" and "aireplay-ng --deauth desired packet length -a attacked router BSSID -c target mac address wlan0". After running these commands, the attack commenced, causing the target device to disconnect from the network due to bandwidth saturation.

For the DOS attack, the Wireshark tool was run on the target device to capture network packets during the attack. As a result of the attack, the target device's bandwidth was filled, causing the device to be unable to perform network operations and eventually disconnect from the network.

To record network traffic during UDP Flood, ICMP Flood, and SYN Flood attacks, the Wireshark tool was started before each attack. During these attacks, UDP, ICMP, and SYN packets were separately sent to the target device. The extensive packets sent led to the saturation of the target system's bandwidth. Due to the bandwidth saturation, the target system was unable to perform network operations, resulting in a service outage.

Before initiating the Man in the Middle attack, Wireshark was run to capture packets on the network. The goal of the attack was to route the target machine's data through the threatening machine before reaching the server. This way, the target machine's network operations pass through the threatening machine before reaching the server.

The CICFlowMeter tool was used to extract features from pcap files obtained with Wireshark for feature extraction. The CICFlowMeter tool was first downloaded and installed. It was then prepared for feature extraction using network traffic data recorded by Wireshark. When starting CICFlowMeter, the data source to be used was specified, and the features to be extracted were selected. The feature extraction process was started by giving the necessary commands to the tool and the feature extraction process required for the classification of pcap files obtained after attacks has been completed. Table 4.2 lists the features extracted by the CICFlowMeter, an open-source tool that generates Bitflows from pcap files and extracts features from these flows [24].

After the feature extraction process of the packets obtained after the attacks was completed, various classification algorithms were used. These algorithms include Naive Bayes, Gradient Boosting, Support Vector Machines, and K-Nearest Neighbors. These algorithms were preferred because they are very fast in terms of calculations and perform well, especially on small data sets. Other models were not preferred because they showed low accuracy and performance. The random forest algorithm was not preferred due to its high memory usage problem, being less understandable, not being successful in data sets containing many features, and being more complex and slower than its alternatives. During the classification process, the necessary packets for the algorithms were first imported.

In the next step, after reading the data set file, a filtering process was performed due to the presence of non-numeric columns in the data. The names of the non-numeric columns are added to the numeric_columns list. Subsequently, only the numeric columns are assigned to the data_numeric variable, creating a data frame containing only numerical columns.

As a next step, the process of separating independent variables (x) and target variables (y) was carried out. The x variable is assigned all columns of the data_numeric data frame except for the last column, while the y variable is assigned the last column of the data_numeric data frame.

After separating the data set (x and y) into training and test sets, the train_test_split function was used to randomly select 80% for training and 20% for testing the x and y data sets. The random_state parameter was used to ensure the randomness of the data set's division or the repeatability of a random process.

The model-building process was carried out for each classification algorithm, and the training process was performed using the training data set (x_train and y_train). The trained model was then used to predict the test data set (x_test).

The prediction results (y_pred) were compared with the actual target values (y_test). The model's accuracy was calculated and printed to the screen, and the necessary data for the comparison process was obtained.

For the Support Vector Machines classification algorithm, the necessary packages were added, and the process of reading the CSV file and filtering non-numeric columns was performed. In the next step, the process of separating independent variables (x) and target variables (y) was carried out. The x variable was assigned all columns of the data_numeric data frame except for the last column, and the y variable was assigned the last column of the data_numeric data frame. Additionally, categorical variables were converted to numerical values.

After separating the data set (x and y) into training and test sets, the train_test_split function was used to randomly select 80% for training and 20% for testing the x and y data sets. The random_state parameter was used for the randomness of the data set's division or the repeatability of a random process.

After the model-building and accuracy score calculation processes, the Support Vector Machines classification process was completed.

Then, the required packages for the Gradient Boosting classification algorithm are imported. In the next step, after reading the CSV file, a filtering process was performed due to the presence of non-numeric columns in the data. The names of the non-numeric columns are added to the numeric_columns list. Subsequently, only the numeric columns are assigned to the data_numeric variable, creating a data frame containing only numerical columns.

The process of separating independent variables (x) and target variables (y) was carried out. The x variable was assigned all columns of the data_numeric data frame except for the last column, while the y variable was assigned the last column of the data_numeric data frame.

After converting categorical variables to numerical values, the data set (x and y) is split into training and test sets. The train_test_split function was used to randomly select 80% for training and 20% for testing the x and y data sets. The random_state parameter was used to ensure the randomness of the data set's division or the repeatability of a random process.

After the model-building and training processes, performance metrics were calculated.

For the K-Nearest Neighbors classification algorithm, the required packages were imported, and the CSV file was read. Due to the presence of non-numeric columns in the data, a filtering process was performed. The names of the non-numeric columns are added to the numeric_columns list. Subsequently, only the numeric columns are assigned to the data_numeric variable. In the next step, the process of separating independent variables (x) and target variables (y) was carried out. The x variable was assigned all columns of the data_numeric data frame except for the last column, and the y variable was assigned the last column of the data_numeric data frame. After converting categorical variables to numerical values, the data set (x and y) was split into training and test sets. The train_test_split function was used to randomly select 80% for training and 20% for testing the x and y data sets. The random_state parameter was used for the randomness of the data set's division or the repeatability of a random process.

The model-building and training processes were completed, and with the completion of the classification processes, the classification algorithm providing the best performance was observed based on the performance values obtained.

3.3. The Other Datasets in the Literature

Various datasets have been created to train IDSs developed to detect intrusions into wireless networks. In this part of our study, we analyze and compare the most commonly used data sets.

3.3.1. KDDCup99

KDDCup99 was created by obtaining data from a military network environment and is one of the most widely used data sets [25][26] . There are 23 attack data and 1 normal class data in the data set. The attack group is categorized as Dos, Probe, R2L and U2R attacks. There are 4898431 training data and 311029 test data in the dataset. 74.41% of the attacks were DoS, 1.33% Probe, 0.07% U2R and 4.68% R2L attacks [27]. KDDCup99 requires less memory and processing power, and the dataset contains easily available features [28]. Although KDDCup99 is widely used, it also has disadvantages. Because it contains synthetic data, it does not match real network traffic. Besides, the amount of training and testing data is huge and therefore has a complex structure. And the detection accuracy rate is low.

3.3.2. NSL-KDD Dataset

To address the shortcomings of the KD99 dataset, The NSL-KDD dataset was introduced by Tavallaee et al. [29]. This dataset was developed by removing unnecessary and redundant data from the KDDCup99 dataset and contains only the data that is truly necessary. There are 37 attacks in total, and 27 attacks were used to test the model and 23 attacks were used to train the dataset. It includes Probe attacks, Denial of Service attacks (DoS), User to Root (U2R) and Remote Local (R2L) attacks. The correct detection rate of NSL-KDD is much higher than that of KDDCup99, and the performance of the models is even higher due to the removal of unnecessary data. However, it is still an improved version of KDDCup99 and since it uses the same data, it does not reflect realistic network data.

3.3.3. UNM Dataset

The UNM dataset was introduced in 2004 and includes data such as buffer overflows, symbolic link attacks and trojan programs [30]. Although UNM is more up-to-date than the KDD dataset, the UNM dataset is extremely limited in coverage and cannot replace the KDD dataset. This data set is not used today. Since it was produced in the 1990s and contains fewer and less complex features compared to modern data sets, it cannot be used by researchers in areas that require in-depth analysis.

3.3.4. CICIDS2017

CICIDS2017 was created by the Canadian Security Institute and includes 5 days of normal and attack data. The data set includes attack types such as DDoS, Brute Force, Botnet, XSS, and SQL Injection. The dataset provides a broad set of features. In this way, researchers are allowed to make in-depth analyses. It contains 3119345 data in total and these data are located in eight different files. Since the data set consists of a lot of data and has a complex structure, it consumes a lot of time for data loading and processing. In addition, there is a large class imbalance in the CICIDS2017 dataset. This causes the intrusion detection system to raise too many false alarms.

3.3.5. UNSW-NB15 Dataset

It started to be developed after it was realized that KDDCup and NSL-KDD did not provide good enough and realistic results in an IDS evaluation [31]. This dataset was created using the IXIA Perfect Storm tool and was created at the Australian Center for Cyber Security (ACCS). Twelve algorithms were used to create the dataset with 49 features, including the class label [32]. The data set consists of attack and normal data, a total of 2.5 million data. It contains Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms attack data. It has more realistic data than its alternatives, but the high amount of data and 49 features can make the modeling process of this data set complicated.

4. Results and Discussions

As a result of the conducted studies, performance metrics have been calculated for each classification. The values related to the data quantity used for the classification process are provided in Table 2. The distribution of data quantities has influenced the interpretation of the obtained performance values. In cases of deauthentication, ICMP, and SYN Flood attacks, the data volume is minimal, leading to classifiers achieving accuracy rates very close to 0 or 1. In our future studies, we aim to increase the data volume associated with these attack types. The amount of data we have in our dataset is given in Table 2.

Table 2. Data Count

Attack	**Amount of Data**
Deauthentication	122
Dos Attack	5430
ICMP Flood	143
SYN Flood	140
Man in the Middle	2161

The accuracy value for the Naive Bayes classification algorithm is 0.74, and the values for other performance metrics are provided in Table 3. Due to the low amount of data associated with Deauthentication, ICMP, and SYN Flood attacks, Naive Bayes has not achieved satisfactory performance in classifying these attacks.

Table 3. Performance Metrics of Naive Bayes Algorithm

Attack	**Precision**	**Recall**	**F1-Score**
Deauthentication	0.00	0.00	0.00
Dos Attack	0.91	0.99	0.95
ICMP Flood	0.00	0.00	0.00
SYN Flood	0.11	0.50	0.18
Man in the Middle	1.00	0.32	0.48

The accuracy value of the Gradient Boosting classification algorithm is 0.91, and other performance metrics are provided in Table 4. The gradient Boosting algorithm has demonstrated better performance compared to the Naive Bayes algorithm in classifying deauthentication and SYN Flood attacks.

Table 4. Performance Metrics of Gradient Boosting Algorithm

Attack	**Precision**	**Recall**	**F1-Score**
Deauthentication	0.33	0.25	0.29
Dos Attack	0.98	0.99	0.99
ICMP Flood	0.00	0.00	0.00
SYN Flood	0.33	0.50	0.40
Man in the Middle	0.91	0.86	0.89

The accuracy value of the Support Vector Machine classification algorithm is 0.84, and other performance metrics are provided in Table 5.

Table 5. Performance Metrics of Support Vector Machine Algorithm

Attack	**Precision**	**Recall**	**F1-Score**
Deauthentication	0.00	0.00	0.00
Dos Attack	0.96	0.86	0.90
ICMP Flood	0.00	0.00	0.00
SYN Flood	0.00	0.00	0.00
Man in the Middle	0.69	0.96	0.80

The accuracy value of the K-Nearest Neighbors classification algorithm is 0.85, and other performance metrics are provided in Table 6.

Table 6. Performance Metrics of K-Nearest Neighbors Algorithm

Attack	**Precision**	**Recall**	**F1-Score**
Deauthentication	0.00	0.00	0.00
Dos Attack	0.93	0.93	0.93
ICMP Flood	0.00	0.00	0.00
SYN Flood	0.00	0.00	0.00
Man in the Middle	0.79	0.82	0.80

The performance values for all classification algorithms are provided in Table 7.

Table 7. Comparison of Performance Values of Classification Algorithms

Algorithm	**Accuracy**	**Precision**	**Recall**
Naive Bayes	0.74	0.40	0.36
Support Vector Machines	0.84	0.51	0.52
Gradient Boosting	0.91	0.33	0.36
K-Nearest Neighbors	0.85	0.44	0.35

As seen in the table above, the highest accuracy value is obtained in the Gradient Boosting classification algorithm, while the lowest is obtained from the Naive Bayes classification algorithm. The accuracy value of the Support Vector Machines classification algorithm is higher than that of the K-Nearest Neighbors classification algorithm.

5. Conclusions and Suggestions

In the created test environment, potential attacks that could threaten wireless networks were simulated. Alongside these cyber threats, network traffic was recorded. The objective was to animate the traffic that might occur during potential cyber threats, record these network packets, and analyze the recorded packets. Different attacks lead to different outcomes in the network, and the varied reactions of the network to these results allow obtaining different data for each attack.

In the next step, the features of the network packets obtained from the created test environment were extracted. These features were prepared for the optimization process and classification steps using machine learning approaches. The CICFlowMeter was used for feature extraction, resulting in different levels of efficiency in the obtained features for each attack. This variation was due to the CICFlowMeter's inability to separate attack packets into their features in some attack scenarios.

For the classification process, various classification algorithms were employed. The scores obtained by the dataset from the applied classification algorithms were collected. The accuracy scores for Naive Bayes, Support Vector Machine, Gradient Boosting, and K-Nearest Neighbors algorithms were 0.74, 0.84, 0.91, and 0.85, respectively. While the scores of K-Nearest Neighbors and Support Vector Machines were very close, there was a significant difference in scores between the highest and lowest-scoring classification algorithms. Although a high accuracy value generally indicates correct classification by the algorithm, the presence of imbalances in classification for cyber attacks suggests considering other performance metrics.

As a result of the performed operations, after the classification processes, the highest score was obtained from the Gradient Boosting classification algorithm, while the lowest score was obtained from the Navie Bayes classification algorithm.

6. Suggestions and Future Works

Since the data amounts of Deauthentication, SYN Flood and ICMP flood packets in our data set are very small, some machine learning classifiers failed to detect the attack data. In the future, we aim to improve the performance of these classifiers by increasing the number of data in our dataset.

The feature extractor used during the feature extraction process for some attacks, CICFlowMeter, has proven to be inadequate as it cannot separate attack packets into their features. In this regard, it is recommended to develop better feature extraction software or conduct a more comprehensive study using existing software with better performance.

Additionally, increasing the number and diversity of cyber attacks and expanding the study using different classification algorithms are suggested. This could lead to obtaining better scores with classification algorithms that perform well under various conditions. Finally, considering the increasing cyber threats with the evolving and developing technology, it is recommended that awareness be raised among the public. Alongside awareness campaigns, increasing scientific studies in this field is also suggested.

References

[1] A. N. Ozalp, Z. Albayrak, and A. Zengin, "Expansion of Wireless Networks using IEEE 802.3af Protocol in Protected Areas," in *5th International Symposium on Innovative Technologies in Engineering and Science*, 2017.

[2] M. Wazid, A. K. Das, V. Chamola, and Y. Park, "Uniting cyber security and machine learning: Advantages, challenges and future research," 2022. doi: 10.1016/j.icte.2022.04.007.

[3] S. GÖNEN, H. İ. ULUS, and E. N. YILMAZ, "Bilişim Alanında İşlenen Suçlar Ve Kişisel Verilerin Korunması," *Bilişim Teknol. Derg.*, vol. 9, no. 3, Sep. 2016, doi: 10.17671/btd.90710.

[4] E. AKBAL, Ş. DOĞAN, T. TUNCER, and N. S. ATALAY, "Adli Bilişim Alanında Ağ Analizi," *Bitlis Eren Üniversitesi Fen Bilim. Derg.*, vol. 8, no. 2, pp. 582–594, 2019, doi: 10.17798/bitlisfen.479303.

[5] K. A. Dhanya, S. Vajipayajula, K. Srinivasan, A. Tibrewal, T. S. Kumar, and T. G. Kumar, "Detection of Network Attacks using Machine Learning and Deep Learning Models," *Procedia Comput. Sci.*, vol. 218, pp. 57–66, 2023, doi: 10.1016/j.procs.2022.12.401.

[6] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues," 2022. doi: 10.3390/s22134730.

[7] A. Mughaid *et al.*, "Improved dropping attacks detecting system in 5g networks using machine learning and deep learning approaches," *Multimed. Tools Appl.*, vol. 82, no. 9, pp. 13973–13995, Apr. 2023, doi: 10.1007/s11042-022-13914-9.

[8] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: principle, practice and challenges," *Artif. Intell. Rev.*, vol. 55, no. 7, pp. 5215–5261, Oct. 2022, doi: 10.1007/s10462-022-10143-2.

[9] D. M. Gezgin and E. Buluş, "Kablosuz Erişim Noktalarına Yapılan DoS Saldırıları," pp. 83–89, 2008.

[10] A. N. Kadhim and S. B. Sadkhan, "Security Threats in Wireless Network Communication-Status, Challenges, and Future Trends," in *2021 International Conference on Advanced Computer Applications (ACA)*, IEEE, Jul. 2021, pp. 176–181. doi: 10.1109/ACA52198.2021.9626810.

[11] D. Cossa, "The Dangers of Deauthentication Attacks in an Increasingly Wireless World," *Iowa State Univ.*, vol. 537, 2014.

[12] R. Cheema, D. Bansal, and S. Sofat, "Deauthentication/Disassociation Attack: Implementation and Security in Wireless Mesh Networks," *Int. J. Comput. Appl.*, vol. 23, no. 7, pp. 7–15, 2011, doi: 10.5120/2901-3801.

[13] W. Liu, "Research on DoS attack and detection programming," in *3rd International Symposium on Intelligent Information Technology Application, IITA 2009*, 2009. doi: 10.1109/IITA.2009.165.

[14] A. N. Ozalp, Z. Albayrak, M. Cakmak, and E. Ozdogan, "Layer-based examination of cyber-attacks in IoT," in *HORA 2022 - 4th International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*, 2022. doi: 10.1109/HORA55278.2022.9800047.

[15] D. Mertkan Gezgin and E. Buluş, "KABLOSUZ AĞLARIN GÜVENLİK AÇIKLARININ EĞİTİM AMAÇLI İNCELENMESİ İÇİN UYGULAMA TASARIMI," *Cilt*, vol. 2, no. 1, pp. 127–135, 2012.

[16] H. (Harshita) Harshita, "Detection and Prevention of ICMP Flood DDOS Attack," *Int. J. New Technol. Res.*, vol. 3, no. 3, p. 263333, 2017, [Online]. Available: https://www.neliti.com/publications/263333/

[17] Z.-Y. Shen, M.-W. Su, Y.-Z. Cai, and M.-H. Tasi, "Mitigating SYN Flooding and UDP Flooding in P4-based SDN," in *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, Sep. 2021, pp. 374–377. doi: 10.23919/APNOMS52696.2021.9562660.

[18] M. Thankappan, H. Rifà-Pous, and C. Garrigues, "Multi-Channel Man-in-the-Middle attacks against protected Wi-Fi networks: A state of the art review," *Expert Syst. Appl.*, vol. 210, p. 118401, Dec. 2022, doi: 10.1016/j.eswa.2022.118401.

[19] B. L. Aylak, O. Oral, and K. Yazici, "Using artificial intelligence and machine learning applications in logistics," 2021. doi: 10.31202/ecjse.776314.

[20] A. N. Özalp and Z. Albayrak, "Detecting Cyber Attacks with High-Frequency Features using Machine Learning Algorithms," *Acta Polytech. Hungarica*, 2022, doi: 10.12700/APH.19.7.2022.7.12.

[21] A. Robles-Velasco, P. Cortés, J. Muñuzuri, and L. Onieva, "Prediction of pipe failures in water supply networks using logistic regression and support vector classification," *Reliab. Eng. Syst. Saf.*, vol. 196, p. 106754, Apr. 2020, doi: 10.1016/j.ress.2019.106754.

[22] V. J. Pandya, "Comparing Handwritten Character Recognition by AdaBoostClassifier and KNeighborsClassifier," in *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, Dec. 2016, pp. 271–274. doi: 10.1109/CICN.2016.59.

[23] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," pp. 1–17, 2020,

[Online]. Available: http://arxiv.org/abs/2008.05756
[24] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *ICISSP 2017 - Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, 2017, pp. 253–262. doi: 10.5220/0006105602530262.
[25] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, L. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: A survey," *Eurasip J. Wirel. Commun. Netw.*, 2013, doi: 10.1186/1687-1499-2013-271.
[26] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," *Comput. Secur.*, 2011, doi: 10.1016/j.cose.2011.08.009.
[27] O. Atilla and E. Hamit, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ*, 2016.
[28] R. Bala, "A REVIEW ON KDD CUP99 AND NSL-KDD DATASET," *Int. J. Adv. Res. Comput. Sci.*, 2019, doi: 10.26483/ijarcs.v10i2.6395.
[29] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009. doi: 10.1109/CISDA.2009.5356528.
[30] Y. Hamid, V. R. Balasaraswathi, L. Journaux, and M. Sugumaran, "Benchmark Datasets for Network Intrusion Detection: A Review," *Int. J. Netw. Secur.*, 2018.
[31] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, 2015. doi: 10.1109/MilCIS.2015.7348942.
[32] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.*, 2016, doi: 10.1080/19393555.2015.1125974.

Author(s) Contributions
İmran KAÇAN: Data collection, data analysis, writing. Batuhan GÜL: Design, writing, review of content, literature review. Fatih ERTAM: Design, Data analysis, technical support, review of content.

Acknowledgments
This study is supported by the Fırat University Scientific Research Projects Coordination Unit with project number TEKF.23.12.

Conflict of Interest Notice
There is no conflict of interest.

Support/Supporting Organizations
This study is supported by the Fırat University Scientific Research Projects Coordination Unit

Ethical Approval and Informed Consent
Our study does not require ethics committee approval.

Availability of data and material
Data will be provided upon request.

Plagiarism Statement
This article has been scanned by iThenticate ™.