E-ISSN 2147-9429

Journal of Polytechnic

POLİTEKNİK DERGİSİ

GAZİ ÜNİVERSİTESİ

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

CLARIVATE ANALYTICS EMERGING SOURCES CITATION INDEX INDEXED

Detection of different windows PE malware using machine learning methods

Makine öğrenimi metotları kullanılarak farklı windows PE kötü amaçlı yazılımların tespiti

Yazar(lar) (Author(s)): *Aynur KOÇAK*[1], *Esra SÖĞÜT*[2], *Mustafa ALKAN*[3], *O. Ayhan ERDEM*[4]

ORCID[1]: *0000-0001-9647-7281*

ORCID[2]: *0000-0002-0051-2271*

ORCID[3]: *0000-0002-9542-8039*

ORCID[4]: *0000-0001-7761-1078*

To cite to this article: Koçak A.,Söğüt E., Alkan M. and Erdem O. A., "Detection of different windows PE malware using machine learning methods ", ***Journal of Polytechnic***, 26(3): 1185-1197, (2023).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Koçak A.,Söğüt E., Alkan M. and Erdem O. A., "Detection of different windows PE malware using machine learning methods ", ***Politeknik Dergisi***, 26(3): 1185-1197, (2023).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1207704

Detection of Different Windows PE Malware Using Machine Learning Methods

Highlights

- *Discussion of different malware attacks and their impact on Windows cyber security.*
- *Testbed preparation called AyEs.*
- *Performing different attacks that will take over the victim system.*
- *Using machine learning methods such as Naive Bayes, J48, BayesNet, IBk, AdaBoost and LogitBoost to detect malware attacks*

Graphical Abstract

According to the information in Figure, there are 2 stages. Stage 1: Normal operating state data and total attack state data of the victim system were obtained. Stage 2: "Attacked state" data was obtained by performing six different attacks separately.

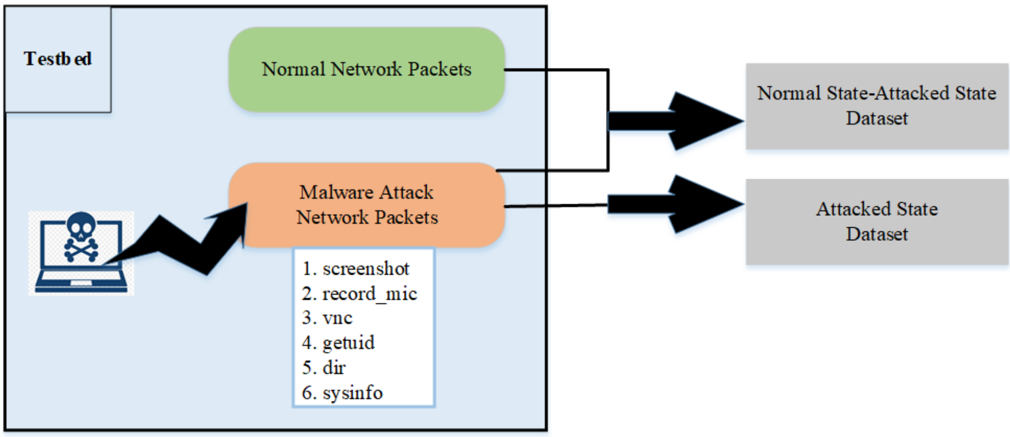


Figure. A general summary of the obtaining the AyEs Dataset

Aim

Detection of malware attacks on Windows systems using machine learning methods.

Design & Methodology

In the study, a testbed was prepared by using virtualization technologies such as VMware Workstation. Malware attacks specific to the vulnerabilities of the Windows system were prepared by using msfvenom and meterpreter tools and these attacks were implemented. Weka tool was used to examine the effects of attacks and to detect attacks. Machine learning methods such as Naive Bayes, J48, BayesNet, IBk, AdaBoost and LogitBoost were used to detect malware attacks.

Originality

Six different malware attacks have been prepared and implemented specifically for Windows systems. Two different datasets were created by collecting the obtained data. While analyzing the datasets, models have been proposed for two different detection systems, whether there is an attack or not and the attack type is determined.

Findings

Our study achieved 98.45% accuracy for the "Normal State-Attacked State" dataset with the J48 algorithm. For the "Attacked State" dataset, it got the best classification result with a success rate of 90.46% using the IBk algorithm.

Conclusion

In our study, contributions are made to the literature by preparing a testbed, obtaining a two-stage dataset, performing two different attack detection processes and providing high performance in attack detection.

Declaration of Ethical Standards

The authors of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Detection of Different Windows PE Malware Using Machine Learning Methods

Araştırma Makalesi / Research Article

Aynur KOÇAK[1], Esra SÖĞÜT[2], Mustafa ALKAN[1], O. Ayhan ERDEM[2],

[1]Faculty Of Technology, Department of Electrical and Electronics Engineering, Gazi University, Turkey
[2] Faculty Of Technology, Department of Computer Engineering, Gazi University, Turkey

(Geliş/Received : 22.11.2022 ; Kabul/Accepted : 03.02.2023 ; Erken Görünüm/Early View : 02.04.2023)

ABSTRACT

The types and application areas of cyber attacks are increasing and diversifying. Accordingly, the effects of attacks are constantly increasing or changing every moment. Among the attacks, malware attacks also have diversified and gained a wide place in the cyber world. With the use of different techniques and methods, there are problems in detecting and preventing malware attacks. These problems cause the systems' cyber security not to be fully ensured. Due to these situations, different malware attacks are discussed in the study, and the effects of attacks on Windows security are examined. A test-bed called AyEs has been prepared. Different attacks have been carried out, such as screenshots, vnc, aimed at hijacking or corrupting the victim system. The AyEs dataset was created by listening to the system network packets obtained due to the attacks. The dataset was preprocessed and made suitable for analysis. Machine learning methods such as Naive Bayes, J48, BayesNet, IBk, AdaBoost and LogitBoost were used on the dataset to detect malware attacks. J48 and IBk methods, which were found to provide high performance as a result of the analyzes, were suggested in the study. In this way, detection systems suitable for possible attack situations against Windows systems will be implemented easily and effectively. In addition to attack detection, an active role will be assumed in determining the type of attack.

Keywords: Dataset, machine learning, malware, testbed, windows system.

Makine Öğrenimi Metotları Kullanılarak Farklı Windows PE Kötü Amaçlı Yazılımların Tespiti

ÖZ

Siber saldırıların türleri ve uygulama alanları çeşitlenerek artmaktadır. Buna bağlı olarak, saldırıların etkileri de her an sürekli artmakta veya değişmektedir. Saldırılar içerisinde malware saldırıları da çeşitlenerek kendisine siber dünyada geniş bir yer edinmiştir. Farklı tekniklerin ve yöntemlerin de kullanılmasıyla malware saldırılarının hem tespiti hem de engellenmesi konularında sorunlar yaşanmaktadır. Bu sorunlar ise sistemlerin siber güvenliğinin tam olarak sağlanamamasına neden olmaktadır. Bu durumlardan dolayı çalışmada farklı malware saldırıları ele alınmış ve saldırıların Windows güvenliği üzerindeki etkileri incelenmiştir. AyEs adı verilen bir test yatağı hazırlanmıştır. Screenshot, vnc gibi kurban sistemi ele geçirmeyi veya bozmayı amaçlayan farklı saldırılar gerçekleştirilmiştir. Saldırılar sonucunda elde edilen sistem ağ paketleri dinlenerek AyEs veri seti oluşturulmuştur. Veri seti önişlemlerden geçirilerek analize uygun hale getirilmiştir. Malware saldırılarının tespiti için veri seti üzerinde Naive Bayes, J48, BayesNet, IBk, AdaBoost ve LogitBoost gibi makine öğrenmesi yöntemleri kullanılmıştır. Yapılan analizler sonucunda yüksek performans sağladığı görülen J48 ve IBk yöntemleri çalışmada önerilmiştir. Bu sayede, Windows sistemlerine yönelik olası saldırı durumlarına uygun olan tespit sistemlerinin kolaylıkla ve etkin şekilde uygulanması sağlanacaktır. Ayrıca saldırı tespitine ek olarak saldırı türü belirlenmesinde de etkin rol üstlenilecektir.

Anahtar Kelimeler: Veri seti, makine öğrenimi, kötücül yazılım, test yatağı, windows sistem.

1. INTRODUCTION

The development and widespread use of the Internet facilitates and accelerates many jobs in the Information Technology world. Unfortunately, these developments also bring with them large-scale security problems. As the importance of information, data and processes increases, possible attacks and possible damages increase. Cyber attacks can have effects that can cause serious financial damage, leaking of confidential information or loss of trust. For these reasons, providing cyber security has become a necessity.

An increasing number of cyber attacks are being carried out against the systems where data is stored or processed, the users using these systems, or the data transmission paths. Attacks are developing, differentiating, and the effects of attacks are increasing. One of these attacks is malware (or malicious software) attacks. These can be different types of attacks, such as ransomware, trojans or viruses.

Malware is a type of malicious software and was created specifically to damage the systems. The malware aims to disrupt system operations and steal sensitive and confidential information. It is a piece of code that can be added, removed or changed in software [1,2]. Malware attacks appear on important issues such as personal

**Sorumlu Yazar (Corresponding Author)*
e-posta : esrasogut@gazi.edu.tr

information, bank account information or e-mail account information. Important information like this can be stolen, modified or deleted by attackers with malware attacks. Malware can infiltrate the system by taking advantage of security vulnerabilities in the network, causing significant damage, especially to institutions and organizations [3]. Therefore, protection from malware attacks is one of the important issues in terms of providing cyber security.

In order to be protected from attacks, the attack must be detected and defined first. The issue of attack detection is taken seriously in the world of cyber security, and many studies are being conducted on this subject. In particular, machine learning, deep learning or artificial intelligence methods are used for malware detection.

In the study, it is aimed to use machine learning to provide cyber security and to detect malware attacks. A testbed was created for this. Here, it is focused on the cyber security vulnerabilities in the local network and how to ensure security. For this, existing Windows 10 security vulnerabilities were used to seize or damage a Windows 10 computer. Portable Executables (PE) files, which are frequently used in systems, can offer convenient and usable ways to implement security threats. Therefore, PE File format was chosen in the study and malware attacks were carried out with this file.

In the testbed, a special malware attack was prepared on the victim system and the attacks were carried out. Detailed analyzes were made by considering the effects of the attacks one by one. The results showed that the victim system was hacked and the attacks were successful. In order to detect the attacks, a new dataset was created by combining the network data that includes the attack types and the network data that does not contain the attack. In addition, the dataset containing the attack types was also evaluated within itself. Machine learning methods have been studied on these datasets and classification processes have been carried out to detect attacks.

When the studies in the literature are examined, it is seen that there are many studies on malware detection. For example, Huang et al. conducted a study using a deep learning method to detect malware for the Windows7 operating system. They used a ready-made dataset to test the proposed malware detection method and were able to detect 94.70% of attacks [4].

Upadhayay et al. combined 3 different datasets in their study. These datasets are Genome, Drebin and Koodous datasets. In their datasets, they listed the permissions given in network traffic as normal and malware. Afterwards, 3 different detection methods were applied to this dataset. These methods are static, dynamic and hybrid detection methods. In addition, machine learning algorithms were applied to each method. The highest accuracy rate of 95.96% was obtained with the Support Vector Machine (SVM) algorithm applied in hybrid detection [5].

Krcal et al. used machine learning method to detect malicious PE files for Windows. One of the PE files datasets provided by Avast was used. Feedforward network method was used and according to the results obtained, 96.0% successful results were obtained for detection [6].

Diaz et al. used the Sophos-ReversingLabs 20 Million Dataset for non-signature-based malware detection for Windows operating systems. A combination of Long Short-Term Memory (LSTM) and LightGBM was used for the classification process. With this method, an accuracy rate of 91.73% was obtained for detection [7].

Mohan et al. aimed to detect malicious software for Windows. The dataset used in the study was created with the Dtrace tool and the feature selection method was applied to the dataset. In this dataset prepared for machine learning use, Decision Tree (DT) and Random Forest (RF) algorithms gave the best accuracy result with 97% [8].

Irshad et al. created a dataset to detect malware in Windows security. After the feature extraction processes were done, the algorithm was applied. Three different algorithms were used for malware detection. Among these, the algorithm with the highest accuracy is RF, with 86.8% [9].

Anderson et al. created the Windows PE malignant and benign files themselves. And they named this dataset EMBER. In their study, MalConv and LightGBM methods were compared and higher accuracy was achieved with LightGBM [10].

In the study, malware attacks against Windows systems were detected by using machine learning methods. There are five main titles in the study. In the first title, there is an introduction and information about the studies done in the related field. In the second title, information about the testbed created and used in the study and the preparation of attacks are given. In the 3rd title, the execution of prepared attacks and obtaining the dataset after the attacks are carried out are explained. Detection of the attacks against the testbed, the results of the analysis and discussion sections are in the 4th title. In the last title, the results of the study and interpretation information are given.

2. TESTBED

A testbed has been prepared for carrying out attacks, monitoring the effects of attacks, observing the state of the victim system and detecting the attack. Simulations have been carried out. While preparing the testbed, different tools and machines were used. VMware Workstation virtualization platform was used and virtual machines were installed on it. Settings have been made to enable them to communicate over the local network. A virtual machine with Windows 10 operating system was used for the victim system, which was the target of the attacks. This machine is allocated 4 GB of RAM and 30 GB of hard disk space. A virtual machine with the Kali

Linux operating system was used for the attacker system that performs the attacks. This machine is allocated 4 GB of RAM and 20 GB of hard disk space.

Before an attack is made, the characteristics of the victim system must be obtained. The IP information of the victim system was obtained by scanning the IP on the local network. Port scanning was also performed in order to access the open ports used by the victim system. After the IP addresses and port information were determined, payload production was carried out using Metasploit frame on the attacker machine.

Payloads with the determined quality are produced with the msfvenom tool. Encoder operations are carried out in order to bypass the created payloads without being caught by security measures. The previously existing msfpayload and msfencode tools are combined with msfvenom.

In this study, PE type payload was generated with msfvenom tool and meterpreter. The payload produced is executable as ".exe" file. Meterpreter is short for Meta-Intepreter and is a high-end payload owned by the Metasploit Framework. There are several reasons for using the Meterpreter payload. Some of those; The meterpreter operates on RAM and does not write to the hard disk. In this way, the victim leaves as little traces on the system as possible. Meterpreter can be developed with various modules without the need for recompilation. In addition, it is quite powerful because it provides communication by dividing into channels. In addition to these, it has advantages such as command history and command completion. “windows/ meterpreter/ reverse_ tcp” is a payload generation code used to gain access to the target system by exploiting security vulnerabilities. Thanks to this payload, Remote File Inclusion security vulnerabilities are used.

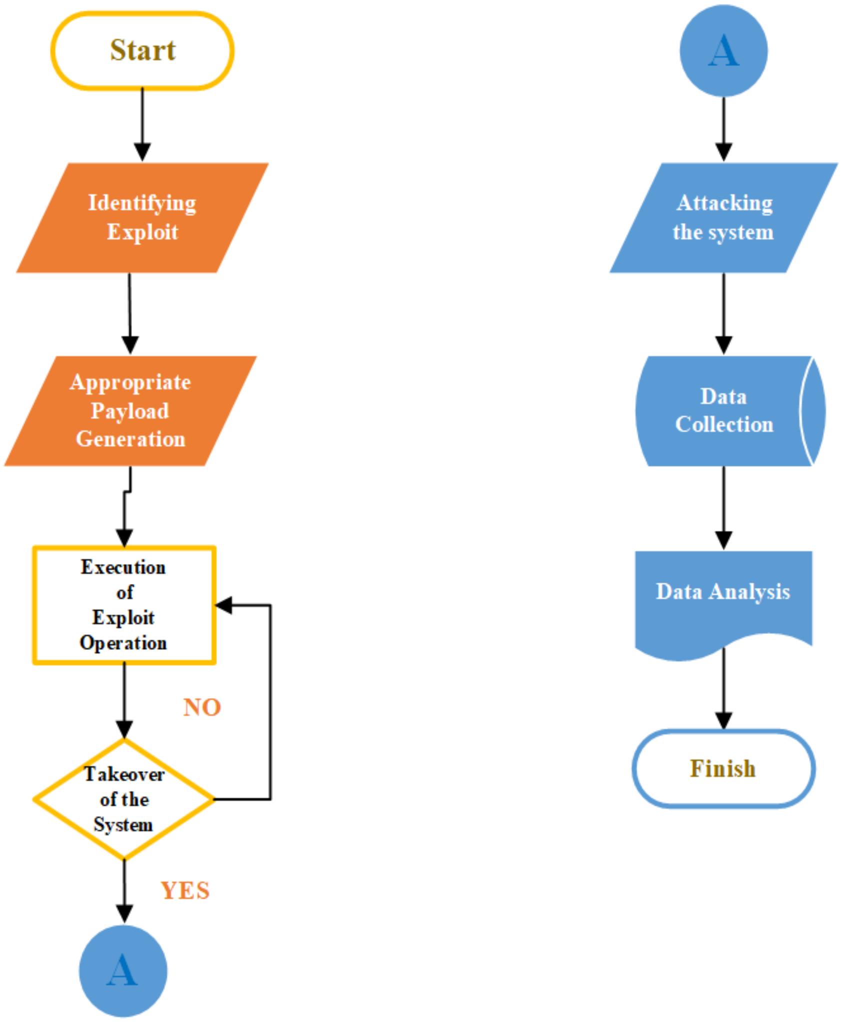


Figure 1. Stages of preparing the attack and performing the exploit

According to Figure 1, exploit detection by exploiting the vulnerabilities of the victim system is the first step in preparing an attack. Then the payload is produced in accordance with the exploit and finally, the exploit is performed. A stager payload is run on the victim system for the attack. In our study, reverse_tcp was chosen as the stager. The selected stager payload creates a Data Definition Language and is injected into another. In the last stage, an encrypted message comes from the target system to Metasploit on the attacker system and the communications flow over encrypted traffic.

3. ATTACKING AND OBTAINING THE DATASETS

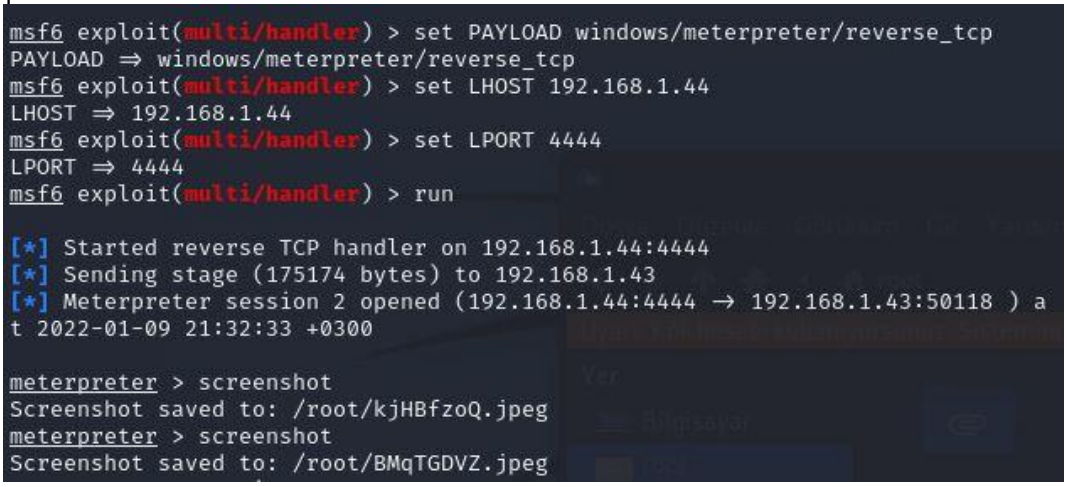
After the payload is created, the necessary settings are made and the target system is accessed. Then, the payload is activated with the "run" command and the target system is captured.

As seen in Figure 2, unauthorized operations were carried out on the victim system. With the example of a "screenshot" attack, operations were performed on the victim system. In the study, six different meterpreter attacks were made and it was aimed at damaging the victim system. Types of attacks:

- Screenshot: It is used to take a snapshot of the victim system.
- Record_mic: It is used to listen to the ambient sound of the victim system for the specified time.
- Vnc: Allows the screen movements of the victim system to be monitored for a certain period of time.
- Getuid: Provides the user name of the victim system to be learned.
- Dir: Displays the current file directory when the victim's system was hacked.
- Sysinfo: Provides victim system information (operating system, number of users, etc.).

Wireshark program was used to listen and collect the network packets of the system where Meterpreter attack examples were applied, separately for each example. Wireshark provides monitoring, analysis and optional filtering of network traffic via a graphical interface [11]. It is one of the most frequently used tools in the literature.

Network flow data is a technology that allows certain parts of the information in the packet to be recorded and analyzed using special algorithms. The features that may be suitable for this study were determined by examining the KDDCUP99 dataset [12]. There are six features determined for the network flow data listened to in the study. In addition to these six columns, there is one class column. This information is shown in Table 1.



```
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.44
LHOST ⇒ 192.168.1.44
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.44:4444
[*] Sending stage (175174 bytes) to 192.168.1.43
[*] Meterpreter session 2 opened (192.168.1.44:4444 → 192.168.1.43:50118 ) a
t 2022-01-09 21:32:33 +0300

meterpreter > screenshot
Screenshot saved to: /root/kjHBfzoQ.jpeg
meterpreter > screenshot
Screenshot saved to: /root/BMqTGDVZ.jpeg
```

Figure 2. Hacking the victim system and carrying out the identified attacks

Table 1. Features and descriptions determined for the dataset

No	Feature	Description
1	Source Port	Port of the computer sending the packet
2	Source IP	IP of the computer sending the packet
3	Destination IP	IP of the computer receiving the packet
4	Destination Port	Port of the computer receiving the packet
5	Protocol	Communication protocol
6	Length	Package size
7	Class	Attack type

The dataset contains the port and IP information of the source machine from which the packet came and the destination machine from which the packet went. In addition, there is the communication protocol used while transmitting the packets and the size of the transmitted packet. The Meterpreter Type column is used to make the classification. This column indicates the attack type and plays an important role in the analysis part.

The data acquisition part of the AyEs dataset prepared in this study is two-stage. Figure 3 shows the process of obtaining the AyEs dataset.

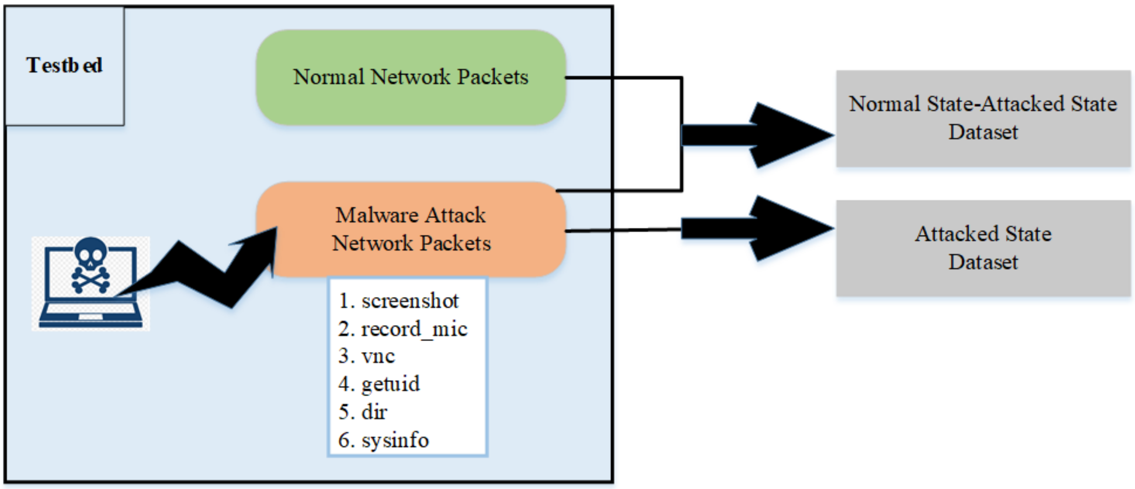


Figure 3. Obtaining the AyEs dataset

According to the information in Figure 3, the stages are as follows:

Stage 1:

- Normal operating state data of the victim system before the attack
- Total attack state data were obtained by performing six different attacks separately

By combining these two datasets, the "normal state-attacked state" dataset was created. In the normal state there are 26856 lines of samples and in the attacked state there are 77184 lines of samples. The total number of samples for these two cases is 104040.

Stage 2:

- "Attacked state" data was obtained by performing six different attacks separately

This "attacked state" dataset consists of 77184 rows of samples. Information about the attacks used in Stage 1 and Stage 2 are given in Table 2.

Table 2. Number of samples according to normal and attacked states in AyEs dataset

No	Attack Type	Number of Samples	Total Number of Samples
1	normal	26856	26856
2	screenshot	9745	77184
3	Record_mic	10480	
4	vnc	12075	
5	getuid	9120	
6	dir	23634	
7	sysinfo	12130	

Table 2 shows the total number of samples obtained by attack types. According to this, when the sample lines consisting of network packets are examined, it is seen that the maximum number of samples belongs to the normal state with no attack, with a rate of 26%. Looking at the attack types, the “dir” attack has the highest number of packet samples with a rate of 31% compared to all attacks. The least applied type of attack is the "getuid" attack, with a rate of 12%.

4. ATTACK DETECTION AND DISCUSSION

Transforming large amounts of data into meaningful information as a result of various analyzes is necessary for intrusion detection systems. There are several methods that achieve this. One of these methods, data mining was used in the study. Weka tool was used to process the obtained datasets, evaluate the data statistically, and draw a meaningful conclusion between the patterns. Weka is a useful tool for analysis operations such as data classification, clustering and regression

[13,14]. It is used to test the performance of many algorithms in cyber attack detection studies.

In this study, the datasets created for the Weka program were converted to ".csv" format. Datasets were primarily preprocessed. Noisy or null data has been cleared. Columns that had no effect on the analysis, such as Time and No, were deleted. Thus, the data is ready to be processed. The steps of the attack detection processes performed for both datasets are given in Figures 4 and 5.

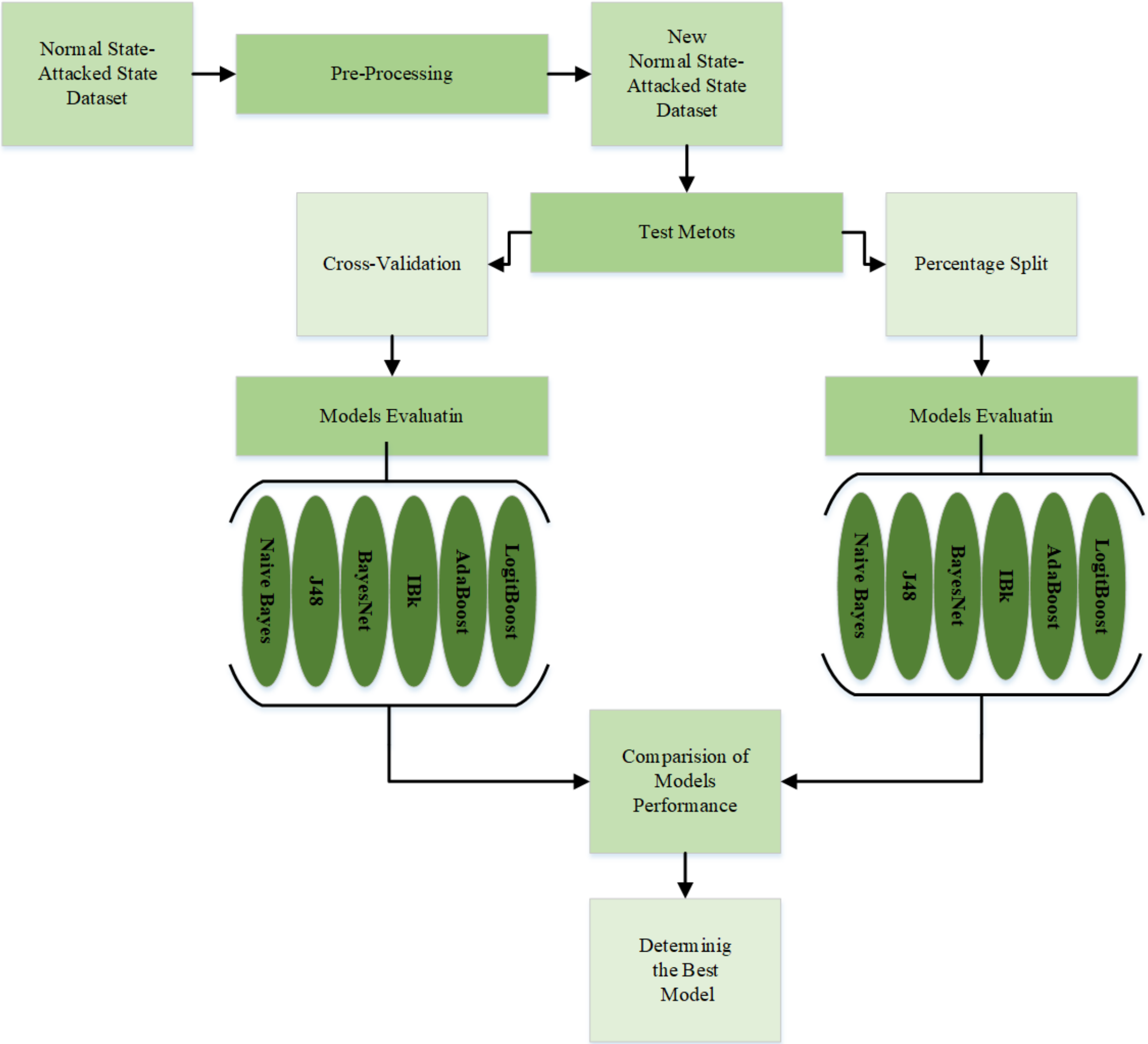


Figure 4. Attack detection stages for 'Normal State-Attacked State' dataset

According to Figure 4 and Figure 5, two different methods were used while creating the model for attack detection with data mining. These are:

- Percentage Split: The data is split in certain proportions. 66% of the data was used for training the model. The remaining data were used for testing purposes.
- Cross Validation: This method is also known as "k-fold cross validation". In this method, the dataset is divided into k equal parts and testing is performed for k different sets. The k value of 10 was chosen in the study.

As shown in Figures 4 and 5, six different data mining algorithms were used to perform attack detection by classification. The obtained results were compared with each other and presented in tables. These algorithms are:

Bayesian Networks: It is an algorithm used to categorize or classify. Probability results are used in the classification process. Naive Bayes model uses a methodology that can achieve high-accuracy results [15]. BayesNet is a successful algorithm for making decisions in uncertain situations. In classification, the data presented for training must have a label class. In this method, probability operations are performed on the training data. The probability values obtained at this stage are used to classify the test data. The formulation of Bayes' theorem is shown in Figure 6.

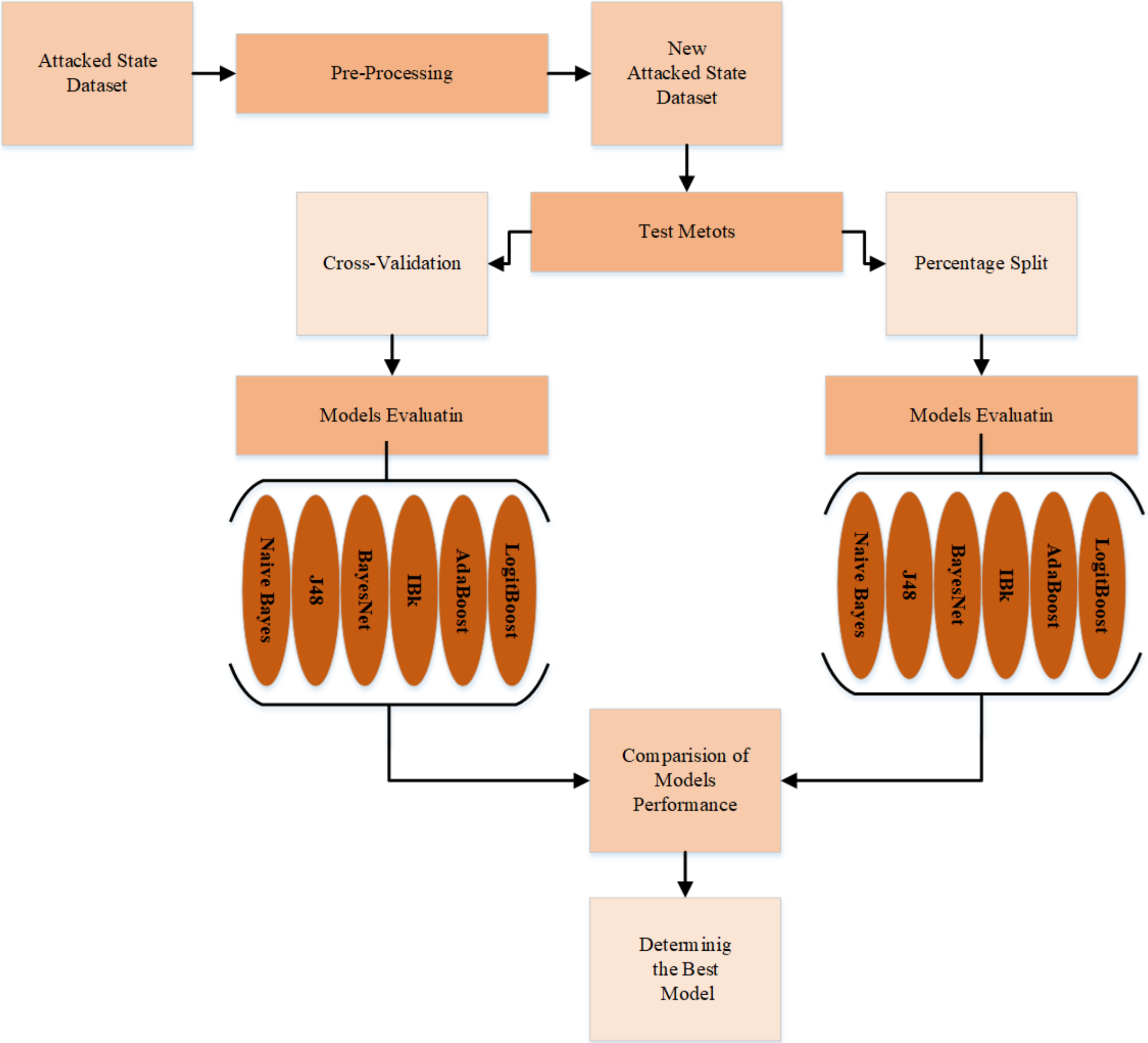
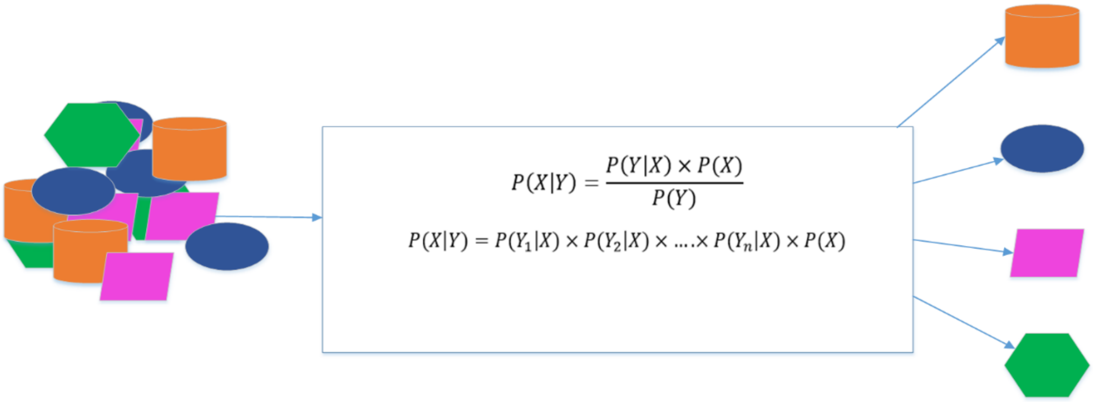


Figure 5. Attack detection stages for 'Attacked State' dataset



$$P(X|Y) = \frac{P(Y|X) \times P(X)}{P(Y)}$$

$$P(X|Y) = P(Y_1|X) \times P(Y_2|X) \times \ldots . \times P(Y_n|X) \times P(X)$$

Figure 6. The working principle Bayes algorithm

For the equation given in Figure 6, the X value indicates which class the classification belongs to. The Y value represents the features in the test data. If the equation is interpreted according to these two variables:

- P(X) value: It is the ratio of the number of samples with X class given in the training set to the total number of samples.
- P(Y) value: It is the ratio of the number of samples with the Y feature in training set to the total number of samples.

- P(Y|X) value: It is the probability that a sample in the X class has the Y feature.
- P(X|Y) value: It is the probability that a sample with feature Y is from class X.

J48: It is a Decision Tree algorithm, and information gain rate is used as the feature selection criterion in this algorithm [16].

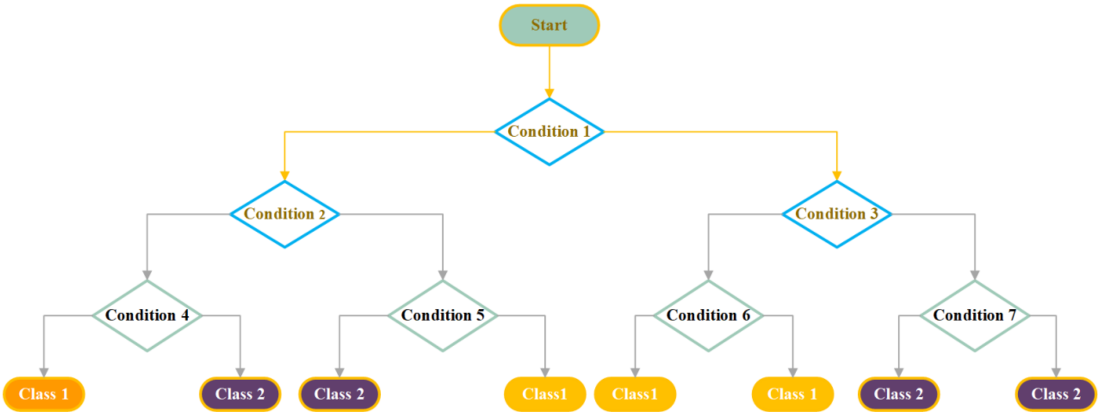


Figure 7. The working principle decision tree algorithm

Decision tree algorithms provide sequential division of the dataset [17]. In order to determine the first condition, the features that are most effective in making the classification are used. And the condition is determined according to these properties. The initial condition is expressed as the root. Sub-conditions are nodes. The last layer, the classification step, is called leaves. The working principle decision tree algoritm is given in Figure 7.

IBk: A K-Nearest Neighbor algorithm that uses the same distance metric. The number of nearest neighbors is a decisive factor for classification. The K number represents the number of samples to be taken from the nearest neighbors. According to the K number, whichever label has more of the nearest neighbors is selected as a result of classification [18]. This situation is illustrated in Figure 8. Different search algorithms can be used to speed up finding the nearest neighbors.

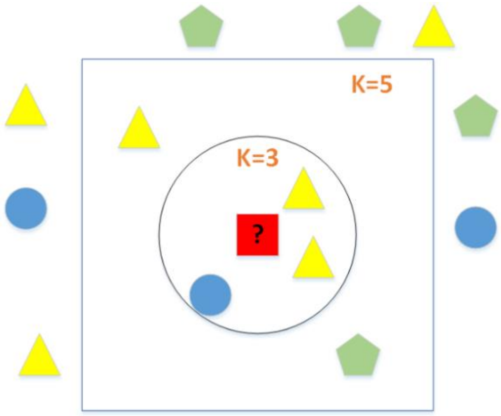


Figure 8. The working principle KNN algorithm

Boosting (AdaBoost and LogitBoost): A group of algorithms that use weighted averages to transform weak learners into strong learners. At the beginning, the label values are separated from each other by applying the 1st iteration. Afterwards, the number of correctly classified samples and the incorrectly classified samples are proportional to each other and a weight value is obtained. Incorrectly estimated tag value is multiplied by this coefficient. Thus, the coefficients of the correct and incorrectly estimated sample numbers are equalized. In the second iteration, the same operations are applied again and the sample values are continued until the correct predicted value is reached. The working logic of iteration states is shown in Figure 9. The boosting algorithm thus aims to place a large number of weak learners in a harmonious order [19].

Algorithms to be used in classification may vary according to the definition of the problem. According to the purpose of the problem, performances can be calculated by calculation accuracy, calculation time or success metrics [20]. In order to measure success in classification models, metrics in the literature are discussed. The analysis results obtained were compared and analyzed according to these metrics. The concept of TP (True Positive) is when the value of the sample in the dataset to be classified is 1, and the classification result is 1. When the actual sample value is 0, and the classification result is 0, it is called TN (True Negative). On the other hand, FP (False Positive) is the case where the classification result of the sample with real value of 0 is 1. Finally, if the classification result of a sample whose real value is labelled 1 is 0, then it is called FN (False Negative). The success metrics used are:

- Accuracy: Indicates how accurately the classification model used predicts all values of the dataset.

- Precision: It is the ratio of correctly predicted positive cases to the sum of TP and FP. A higher ratio indicates the precision of the classification model is better [21].
- Recall: This concept is equal to the ratio of the TP states to the sum of the TP and FN states.
- F-Measure: It is a metric that considers the harmonic mean of the precision and recall metrics.
- TP RATE: It is the ratio of the TP status to the sum of cases that actually have a sample value of 1.
- FP RATE: It is the ratio of the FP status to the sum of cases that actually have a sample value of 0 [22].
- ROC AREA: Shows the accuracy of normal and attack classification for our study. The closer its value is to 1, the higher the performance of the classification method.

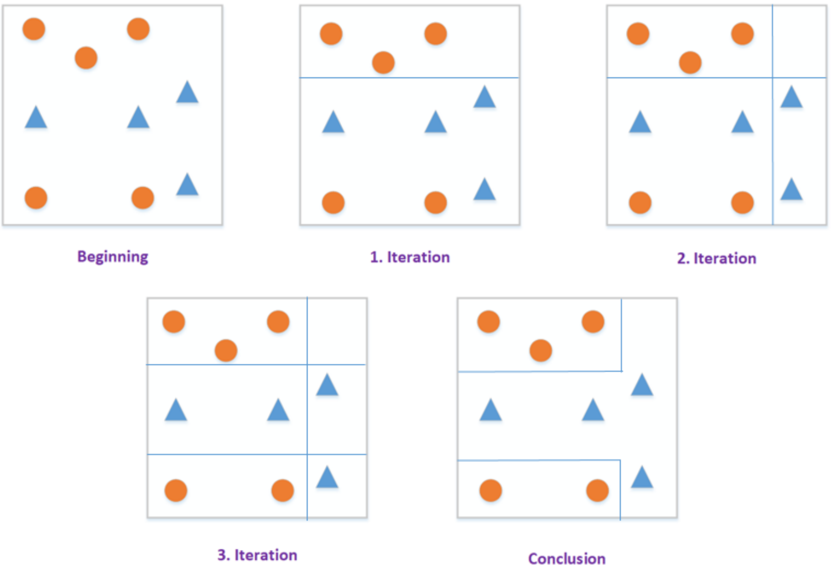


Figure 9. The working principle Boosting algorithm

4.1. Data Analysis for Normal State-Attacked State

The normal state data recorded without the attack and the attacked state data obtained after applying six attack types were combined in one place. According to this dataset obtained, the analysis of whether there is an attack on the Windows system has been made. The results obtained are given in Table 3.

According to Table 3, six different algorithms were used to classify for attack or normal states. Two different test methods, Cross-Validation and Percentage Split were applied for each. Accordingly, the algorithm with the highest classification success rate (accuracy) is J48 with the Cross-Validation method. The J48 algorithm gave the best results in determining whether there was an attack on the obtained dataset. The time spent by the algorithm for analysis is above the average. AdaBoost algorithm showed the worst performance in attack detection. The success rate for this algorithm is low for both methods.

The best results obtained with the success metrics in the classification models are given in Table 4. Considering the Precision, Recall, F-Measure, TP Rate, and ROC Area metrics, the highest values were obtained with the J48 algorithm. Considering the time spent doing classification in the Weka program, the IBk algorithm performed best for the first method and worst for the second method. The J48 algorithm worked longer than the other algorithms except for the IBk. Furthermore, the IBk algorithm performed the best classification after the J48 algorithm. Looking at the FP Rate, the BayesNet algorithm gave the best results.

4.2. Data Analysis for Attacked State

After applying six different attack types separately, the data obtained from the situations were combined in one place. According to this dataset, it has been determined whether the attack belongs to which type. The results obtained are given in Table 5.

According to Table 5, six different algorithms were applied to classify the attacks and 2 different test methods were applied for each. For both methods used in the analysis, the IBk algorithm had a higher accuracy rate than other algorithms. Looking at the time taken for analysis, different lengths were observed. AdaBoost algorithm obtained the lowest accuracy value for both methods. The running time of this algorithm is close to the average.

When the result of the success metrics in Table 6 were examined, the highest rates were obtained for Precision, Recall, F-Measure, and TP Rate with the IBk algorithm. Considering the time the algorithms work for analysis, the algorithms that finish classification in the shortest time are IBk and Naive Bayes. The longest analysis is made by the Percentage-Split method of the IBk algorithm. BayesNet for FP Rate and J48 for ROC Area gave the best results. In addition, the J48 algorithm made the best classification after the IBk algorithm.

Table 3. Data analysis results for 'Normal State-Attacked State'

No	**Algorithms**	**Test Methods**	**Number of Correctly Classified**	**Number of Incorrectly Classified**	**Accuracy**	**Calculation Time**	**Precision**	**Recall**	**F-Measure**	**TP Rate**	**FP Rate**	**ROC Area**
1	**Naive Bayes**	Cross-Validation (10)	99381	4394	95.765	0.05sn	0.963	0.958	0.959	0.958	0.016	0.996
		Percentage Split (%66)	33647	1636	95.363	0.10sn	0.960	0.954	0.955	0.954	0.019	0.994
2	**J48**	Cross-Validation (10)	102170	1605	**98.453**	23.22sn	0.985	0.985	0.984	0.985	0.041	0.999
		Percentage Split (%66)	34693	590	**98.327**	26.99sn	0.983	0.983	0.983	0.983	0.043	0.998
3	**BayesNet**	Cross-Validation (10)	99671	4104	96.045	0.19sn	0.966	0.96	0.961	0.96	0.014	0.997
		Percentage Split (%66)	33734	1549	95.609	0.16sn	0.962	0.956	0.957	0.956	0.016	0.996
4	**IBk**	Cross-Validation (10)	101538	2237	97.844	0.01sn	0.978	0.978	0.978	0.978	0.047	0.998
		Percentage Split (%66)	34422	861	97.557	249.9sn	0.976	0.976	0.975	0.976	0.05	0.996
5	**AdaBoost**	Cross-Validation (10)	80022	23753	77.111	1.19sn	0.814	0.771	0.698	0.771	0.652	0.719
		Percentage Split (%66)	27096	8187	76.796	0.99sn	0.812	0.768	0.694	0.768	0.648	0.72
6	**LogitBoost**	Cross-Validation (10)	83671	20104	80.627	0.93sn	0.841	0.806	0.761	0.806	0.551	0.828
		Percentage Split (%66)	28359	6924	80.375	0.58sn	0.840	0.804	0.758	0.804	0.547	0.829

Table 4. Data analysis success metrics for 'Normal State-Attacked State'

	Accuracy	**Calculation Time**	**Precision**	**Recall**	**F-Measure**	**TP Rate**	**FP Rate**	**ROC Area**
Algorithms	J48	IBk	J48	J48	J48	J48	BayesNet	J48

Table 5. Data analysis results for 'Attacked State'

No	Algorithms	Test Methods	Number of Correctly Classified Instances	Number of Incorrectly Classified Instances	Accuracy	Calculation Time	Precision	Recall	F-Measure	TP Rate	FP Rate	ROC Area
1	**Naive Bayes**	Cross-Validation (10)	67110	9809	87.247	0.05sn	0.891	0.872	0.874	0.872	0.200	0.985
		Percentage Split (%66)	22671	3481	86.689	0.21sn	0.884	0.867	0.869	0.867	0.021	0.983
2	**J48**	Cross-Validation (10)	68621	8298	89.212	16.63sn	0.898	0.892	0.888	0.892	0.036	0.992
		Percentage Split (%66)	23218	2934	88.781	0.14sn	0.888	0.888	0.887	0.888	0.031	0.992
3	**BayesNet**	Cross-Validation (10)	67538	9381	87.804	0.31sn	0.897	0.878	0.88	0.878	0.018	0.987
		Percentage Split (%66)	22866	3286	87.435	0.25sn	0.893	0.874	0.876	0.874	0.019	0.986
4	**IBk**	Cross-Validation (10)	69585	7334	**90.465**	0.05sn	0.904	0.905	0.904	0.905	0.025	0.989
		Percentage Split (%66)	23474	2678	**89.759**	97.6sn	0.897	0.898	0.897	0.898	0.026	0.988
5	**AdaBoost**	Cross-Validation (10)	29969	46950	38.961	0.28sn	-	0.390	-	0.390	0.291	0.565
		Percentage Split (%66)	10116	16036	38.681	0.11sn	-	0.387	-	0.387	0.286	0.568
6	**LogitBoost**	Cross-Validation (10)	47852	29067	62.210	2.15sn	0.740	0.622	0.602	0.622	0.164	0.887
		Percentage Split (%66)	16271	9881	62.217	2.40sn	0.741	0.622	0.606	0.622	0.162	0.885

Table 6. Data Analysis Success Metrics for 'Attacked State'

	Accuracy	Calculation Time	Precision	Recall	F-Measure	TP Rate	FP Rate	ROC Area
Algorithms	IBk	IBk, Naive Bayes	IBk	IBk	IBk	IBk	BayesNet	J48

4.3. Discussion

Studies that have been done before and examined in the introduction are placed in Table 7. These studies and our own work are compared in this section. Analysis results are handled using the Accuracy metric, which is frequently used in the literature.

Table 7. Comparison of studies

References	Algorithms	Accuracy (%)
[4]	VGG16 network hybrid visualization	94.70
[5]	RF, SVM	95.96
[6]	Feedforward network	96.00
[7]	LSTM+LightGBM	91.73
[8]	RF, DT	97.00
[9]	RF	86.80
[10]	LightGBM	98.20
Our Study	J48	98.45
	IBk	90.47

According to the results given in Table 7, different methods were used to detect malware attacks on Windows systems. Unlike the studies examined, in our study, six different algorithms were used together with two different test methods. As a result of the analysis for attack detection, the highest success rates were obtained with the IBk and J48 algorithms. In our study, there are two stages for attack detection. In the first stage, a higher attack detection success rate was achieved with the J48 algorithm than the studies in the literature. In the second stage, the highest success value was obtained with the IBk algorithm in detecting the attack type, but a lower rate was achieved compared to the literature. In our study, contributions are made to the literature by preparing a testbed, obtaining a two-stage dataset, and providing high performance in malware detection processes.

5. CONCLUSION

In information technologies, malware attacks on Windows systems can cause serious problems. In order to prevent possible damage, it has become necessary to provide cyber security for Windows systems. For this, first of all, it is necessary to determine the attack types and to detect the attack accordingly. There are many studies carried out for this purpose and using different methods. In order to contribute to the literature in this field and to give a different perspective, a study on intrusion detection has been carried out.

In addition to the studies in the literature, a special testbed was prepared and named as AyEs. Certain attacks have been carried out against this testbed and two different datasets have been obtained. Analyzes were made using six different algorithms for Normal State-Attacked State and Attacked State datasets. And two different test methods were applied for each algorithm. In the Normal State-Attacked State analysis, a classification is made as to whether it is an attack or a normal state. In the Attacked State analysis, six different attacks were classified.

According to the analysis results obtained; For the Normal State-Attacked State dataset, an accuracy rate of 98.45% was obtained with the Cross-Validation method. For this, the best classification was made with the J48 algorithm. In the Attacked State dataset, among the six algorithms, the IBk algorithm provided the highest performance with the Cross-Validation method. This algorithm gave the best classification result with a success rate of 90.46%.

For future studies, a more comprehensive experimental environment can be prepared by developing the testbed and technologies we use here. In addition, different types of attacks can be applied to this testbed. Different machine learning or deep learning methods can be used for analysis and detection of attacks. Different studies can be done to contribute to the cyber security of Windows systems.

DECLARATION OF ETHICAL STANDARDS

The authors of this article declare that the materials and methods used in their studies do not require ethical committee approval and legal-specific permission.

AUTHORS’ CONTRIBUTIONS

Aynur KOÇAK: Performed the experiments, analyse the results and wrote the manuscript.

Esra SÖĞÜT: Analyse the results and wrote the manuscript.

Mustafa ALKAN: Wrote the manuscript.

O. Ayhan ERDEM: Wrote the manuscript.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] Mithal, T., Kshitij S., and Dushyant K. S., ”Case studies on intelligent approaches for static malware analysis”, ***Emerging Research in Computing, Information, Communication and Applications,*** Springer, Singapore, 555-567, (2016).

[2] Vatamanu, C., et al., ”A comparative study of malware detection techniques using machine learning methods”, ***Int. J. Comput. Electr. Autom. Control Inf. Eng***., 555-567, (2016).

[3] Al-Janabi, M., and Altamimi, A. M., "A Comparative Analysis of Machine Learning Techniques for Classification and Detection of Malware," ***The 21st International Arab Conference on Information Technology***, 1-9, (2020).

[4] Huang, X., Ma, L., Yang, W. et al., “A Method for Windows Malware Detection Based on Deep Learning”, ***J Sign Process Syst***, 93, 265–273, (2021).

[5] Upadhayay, M., Sharma, A., Garg, G., and Arora, A., "RPNDroid: Android Malware Detection using Ranked Permissions and Network Traffic", ***The Fifth World Conference on Smart Trends in Systems Security and Sustainability***, 19-24, (2021).

[6] Krcal, M., Svec, O., Balek, M., and Jasek, O,. “Deep convolutional malware classifiers can learn from raw executables and labels only*”,* ***International Conference on Learning Representations Workshop Track***, (2018).

[7] Diaz, J. A., and Bandala, A., "Portable Executable Malware Classifier Using Long Short Term Memory and Sophos-ReversingLabs 20 Million Dataset", ***TENCON 2021 - 2021 IEEE Region 10 Conference***, 881-884, (2021).

[8] KP. A. M., Chandran, S., Gressel, G., Arjun, T. U., and Pavithran, V., "Using Dtrace for Machine Learning Solutions in Malware Detection", ***The 11th International Conference on Computing, Communication and Networking Technologies***, 1-7, IEEE, (2020).

[9] Irshad, A., Maurya, R., Dutta, M. K., Burget, R., and Uher, V., “Feature optimization for run time analysis of malware in windows operating system using machine learning approach”, ***The 42nd International Conference on Telecommunications and Signal Processing***, 255-260, IEEE, (2019).

[10] Anderson, H., and Roth, P., “EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models”, 2018, ***ArXiv***, abs/1804.04637.

[11] **Internet**: Wireshark, www.wireshark.org.

[12] **Internet**: “KDD Cup 1999 Data”, kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[13] **Internet**: “Weka 3: Machine Learning Software in Java”, https://www.cs.waikato.ac.nz/ml/weka/.

[14] Söğüt, E. & Erdem, O. A., Endüstriyel Kontrol Sistemlerine (SCADA) Yönelik Siber Terör Saldırı Analizi. ***Politeknik Dergisi***, 23 (2), 557-566, (2020).

[15] Choudhary, S., and Sharma, A., "Malware Detection & Classification using Machine Learning", ***International Conference on Emerging Trends in Communication, Control and Computing***, 1-4, (2020).

[16] Quinlan, J. R., “Induction of Decision Trees”, ***Machine learning***, 1(1), 81-106, (1986).

[17] Kasım, Ö., “Malicious xss code detection with decision tree”. ***Journal of Polytechnic***, 23 (1), 67-72, (2020).

[18] Türkoğlu, M., Polat, H., Koçak, C., and Polat, O., “Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection”, ***Expert Systems with Applications***, 203, (2022).

[19] Nahar, N., Ara, F., Neloy, M. A. I., Barua, V., Hossain, M. S., and Andersson, K., "A Comparative Analysis of the Ensemble Method for Liver Disease Prediction", ***The 2nd International Conference on Innovation in Engineering and Technology***, 1-6, (2019).

[20] Koç, K. , Demirtaş, M. & Çetinbaş, İ., Parameter “Extraction of Photovoltaic Models by Honey Badger algorithm and Wild Horse Optimizer”. ***Journal of Polytechnic***, (Erken Görünüm), (2023).

[21] Oduro, M. S., Yu, H., and Huang, H., "Predicting the Entrepreneurial Success of Crowdfunding Campaigns Using Model-Based Machine Learning Methods", ***The International Journal of Crowd Science***, 6(1), 7-16, (2022).

[22] Hashim, A. S., Awadh, W. A., and Hamoud, A. K., “Student performance prediction model based on supervised machine learning algorithms”, ***IOP Conference Series: Materials Science and Engineering***, 928(3), 032019, IOP Publishing, (2020).