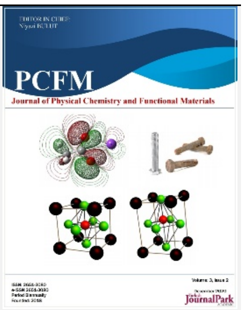
Journal of Physical Chemistry and Functional Materials

Home Page of Journal: https://dergipark.org.tr/jphcfum



A Hybrid Method Based On A Genetic Algorithm That Uses Network Packets To Classify Spyware

Irfan Kilic[a]*, Orhan Yaman[b] , Edanur Erdogan[b], Melisa İrem Aslan[b]

[a]Firat University, Department of Information Technologies, Elazig, Turkey
[b]Firat University, Department of Digital Forensic Engineering, Elazig, Turkey
** Corresponding author: E-mail: irfankilic@firat.edu.tr*

ABSTRACT

The emergence of the Internet has led to the emergence of cyber-attacks and malware. Malware installed on mobile devices, including computers, phones, and tablets, can be used by attackers to access users' data. This study aims to use decision trees (DT) and genetic algorithms (GA) using a meta-heuristic approach to detect spyware, a category of malware, by analyzing network packets in a Windows operating system environment. When the literature is examined, it is noteworthy that there is a lack of studies on the detection of spyware using network packets. This situation was the driving force for this study. In order to carry out the study, an experimental environment was created by utilizing the laboratory facilities of Firat University, Faculty of Technology, Department of Forensic Informatics Engineering. In this experimental environment, various network packets were collected using different spyware applications. The data set was subjected to feature extraction using Tshark software. The effectiveness of meta-heuristics compared to the mathematical method of neighborhood component analysis (NCA) is demonstrated on the benchmark dataset. Therefore, a genetic algorithm (GA) was used to select the most weighted features among the extracted features. The selected features were classified with the decision tree (DT) algorithm. The results obtained are at the desired level for future studies.

ARTICLE INFO

Keywords:
Spyware classification
Genetic algorithm
Decision trees
Neighborhood components analysis
Network packets

Received: 2024-11-05
Accepted: 2024-11-10
ISSN: 2651-3080
DOI: 10.54565/jphcfum.1579687

1. Introduction

Nowadays, with the rapid development of the internet and technology, viruses and malware are constantly evolving to harm users. For this reason, the security of user data has become very important. By taking advantage of system vulnerabilities with malicious software, important data of users and institutions can be captured, changed, or used for malicious purposes. Malware; There are many types such as ransomware, spyware, worms, trojan horses, adware, and fileless malware. The main types of malware are shown in Figure 1 [1]. Spyware, which is a common method of cyber-attacks, is software that is used by secretly infecting computers and other technological devices to infiltrate users' computer systems, monitor their activities, and inform the cyber-criminal. Additionally, spyware can masquerade as a program of your operating system and continue to run in the background. The actions performed by the spyware are shown in Figure 2.

From a cyber security perspective, it is very important to quickly and accurately identify spyware on the Windows operating system, which is most commonly used on desktop computers. Spyware can be installed and hidden on all kinds of technological devices. At the same time, such software can perform actions such as listening to the keyboard (keylogger), taking application and desktop screenshots, monitoring network connections, browsing activity, that is, accessing the websites you visit on your device, accessing your e-mails, changing or disabling the settings of your device. Nowadays, many spywares can be found in antivirus software or package tracking programs. However, since some spyware programs are not trojans or viruses, they may be difficult to detect with antivirus software, or searching for such programs may take too long and be too costly.

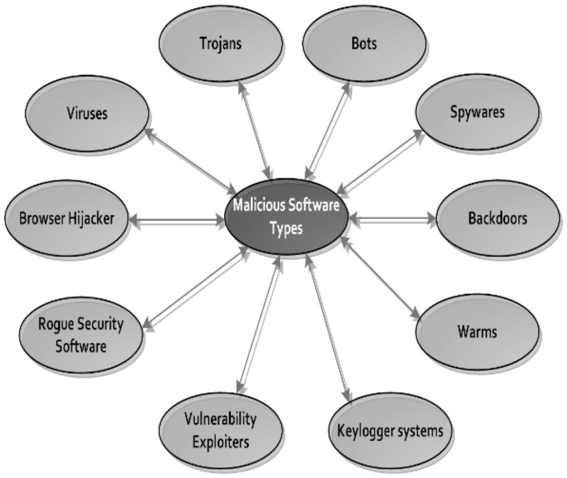


Figure 1 Main types of malware

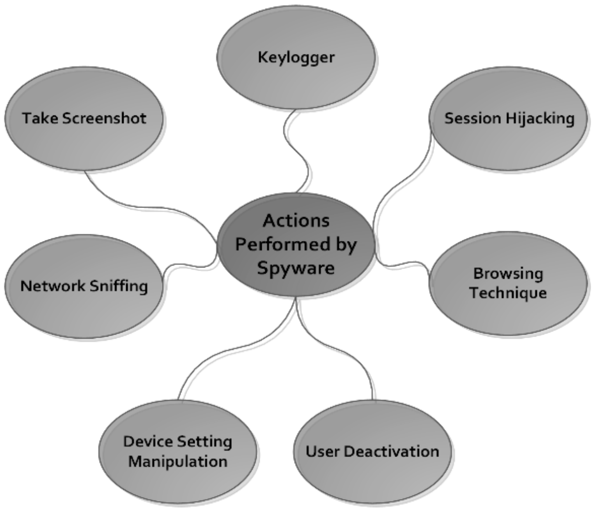


Figure 2 Actions performed by the spyware

Spyware can be installed and hidden on all kinds of technological devices. At the same time, such software can perform actions such as listening to the keyboard (keylogger), taking application and desktop screenshots, monitoring network connections, browsing activity, that is, accessing the websites you visit on your device, accessing your e-mails, changing or disabling the settings of your device. Nowadays, many spywares can be found in antivirus software or package tracking programs.

However, since some spyware programs are not trojans or viruses, they may be difficult to detect with antivirus software, or searching for such programs may take too long and be too costly.

Spyware Types;

- Adware: This type of software is a type of malicious software that broadcasts advertisements on the computer screen through the browser. It runs in the background of computers, tablets, or mobile devices.
- Keyboard Loggers: Keylogger software records keyboard keystrokes on the computer. This type of software runs in the background while using the mobile device and monitors all operations performed on the keyboard.
- Trojan Horses: Trojans, which appear innocent to users, are spyware programs used for malicious purposes. Trojan horses can be disguised by attackers as files such as software, music, or video.
- Mobile Spyware: Mobile spyware monitors network traffic and records information without the user's knowledge and permission.
- Rootkits: Attackers can access and remotely control users' computers with Rootkits software.
- Browser Hijackers: Browser hijackers are widely used spyware that can change the settings of users' browsers.
- Infostealers: This type of spyware runs in the background and infects personal computers, servers, etc. It collects confidential data such as logins and passwords on devices.
- Red Shell: This malware is installed on the computer and used to monitor web-based activities.

Literature Background

There are many studies to detect malware on Windows and Android operating systems [1], [2], [3]. A summary of the studies conducted in the literature is presented in Table 1.

Table 1 Summary of studies in the literature

Reference, Year	Platform	Explanation	Method	Results
Bulut et al. [4], 2017	Android/Windows	Malware detection	Denoising autoencoder + Multi-layer perceptron	Pre: 92.5% Rec: 92.5% F1: 92.3% Acc: 93.67%
Dinçer and Doğru [5], 2017	Android	Malware detection	Survey work	Survey
Utku [6], 2022	Android	Malware detection with network traffic analysis	LSTM	Pre: 94.5% Rec: 97.4%

				F1: 95.9% Acc: 95.0%
Mehtab et al. [7], 2020	Android	Malware analysis	Machine learning, AdDroid	Pre: 99.33% Rec: 99.36% Acc: 99.11%
Pektaş and Acarman [8], 2020	Android	Malware detection	Deep Learning, SDNE32	Pre: 98.84% Rec: 98.47% F1: 98.65% Acc: 98.86%
Bakour and Ünver [9], 2021	Android	Malware detection	Deep Learning, DipVisDroid	Acc: 98.96%
Tokmak and Küçüksille [10], 2019	Windows	Malicious Windows executables	Feature Extract: PCA Deep Learning	Acc: 100% for PCA 150
Bauri et al. [11], 2022	Windows	Windows Post Exploitation	Survey work	Survey
Yadav and Randale [3], 2015	-	Keylogger spyware attack detection and prevention	Honeypot and Encryption Algorithm	No data
Javaheri et al. [12], 2018	-	Detection of spyware and ransomware	Decision tree	Acc: 92.32%
NarasimaMallikarajunan. K. et al [13],	-	Spyware Detection	-	No data
Dama [14]	Windows	Spyware design	-	No data
Erginay [15]	-	Anomaly detection in network traffic	Machine learning	(Accuracy) ANN: 99.4% SVM: 98.6% kNN: 99.8% DT: 99.9% NB: 98.9%
McLaren et al. [16]	Windows	Detection of malware	Ruleset derived from Decision Tree	FPR (False Positive Rate): 1.93% R_{bad}:

When the studies in the literature are examined, malware has been detected mostly for Windows and Android operating systems. According to Türkiye malware environment statistics;

- 97% of mobile malware affects the Android operating system,
- 96% of malware attacks target Microsoft Windows operating systems [4].

In the literature, malware detection is generally done using machine learning and deep learning-based methods. In this study, it is aimed to use network packets to detect spyware, which is a type of malicious software. Network data obtained from different scenarios on Windows computers will be used. With the machine learning-based methods to be developed, adware, keyboard loggers, Trojan horses, rootkits, browser hijackers, and information thieves can be detected. The unique aspect of the project is the ability to detect real-time spyware using network packets. Thus, spyware detection can be done simultaneously even on more than one Windows computer on the network.

Nowadays, shopping, education, health, etc. Along with important work, activities such as social networks, internet games, and news follow-up are carried out over the internet, which further attracts users to the cyber world. As users turn to such areas, it causes an increase in attacks on the internet by cyber criminals. In addition to the increase in attacks, there is not the same level of awareness among users and institutions/companies providing IT services. This awareness must be created by institutions and companies providing IT services and the necessary precautions must be taken [14].

Spyware poses the following risks;

- Obtaining users' username and password information,
- Installation of advertising or malicious software beyond the user's control [17],
- Obtaining users' bank and credit card information,
- Unauthorized seizure of documents on computers,
- Capture of personal data such as e-mail and calendar,
- Seizure of personal photo and video files,
- System components such as data transmission capacity and operations can be used by cybercriminals without the user's knowledge (robot network, etc.),

Problem Definiton, Motivation, and Contributions

- Slowdown of your computer and network performance due to spyware running in the background of the computer,
- Legal sanctions may occur due to behaviors that may constitute a crime due to transactions made without the consent of the users,
- If spyware installed on the operating systems of Internet banking used on mobile devices cannot be prevented, money will be withdrawn from people's bank accounts,
- Risk of leakage of documents such as conversations, e-mails, and important correspondence made by corporate companies and military and public institutions (military espionage),
- It is a risk that mobile devices with computer features such as computers, phones, smartwatches, and tablets, which are found everywhere today, contain spyware and programs such as voice recording and hidden cameras that secretly monitor or listen to users.

Malware detection methods are divided into two main parts: static analysis and dynamic analysis, as seen in Figure 3 [4]. Static analysis is the rapid collection of information about spyware on the target computer without running it. Dynamic analysis is the analysis process performed on the information obtained by running the spy software on the computer.

In this study, it was tried to show the effectiveness of the Genetic Algorithm (GA), which forms the basis of meta-heuristic improvement methods according to computational methods within the scope of dynamic analysis, on a benchmark data set as a feature selector.

Static analysis

Static analysis techniques can be easily applied if the source codes of the software to be analyzed can be found. But usually, no information can be found about the source codes of the software to be analyzed, only a binary file is available. Thanks to utilities (VirusTotal, Dependency Walker, etc.), you can view URLs, IP addresses, etc. By obtaining information, you can learn some information about whether this software is harmful or not. However, the information obtained through static analysis may not give accurate results. For this reason, dynamic analysis should be performed together with the static analysis method 4.

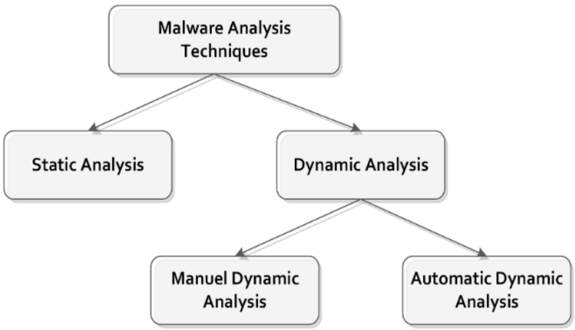


Figure 3 Malware analysis methods

Dynamic analysis

It is the analysis of the software to be analyzed by running it in a virtualized environment (virtual machine or sandbox) and tracking and analyzing all statuses of the environment (file systems, registry, transaction status, IP detection, whois information detection, etc.). As a result of the analysis, file-directory movements, IP traffic, and internet activities on the user's computer are obtained while the software is running [4].

With our study;

- Genetic Algorithm was used for feature selection.
- The features selected with GA were classified using Decision Trees (DT).
- In addition, Neighborhood Component Analysis (NCA), a mathematical method for feature selection, was used and compared with GA.
- The feasibility of a lightweight classifier with GA feature selection was demonstrated.
- The aim here is to show the usability of meta-heuristic methods such as GA for feature selection instead of mathematical methods rather than high accuracy.

It is aimed to prevent problems occurring on computers with spyware (slowdown in the computer system, etc.) and to implement it by using the network and file system, which is a dynamic analysis method.

As study motivation;

- Detecting and classification spyware using network packets
- To be able to select the most meaningful parameters from the packets collected from the Wireshark program.
- The objective of this study is to evaluate the usability of an effective feature selector in conjunction with the meta-heuristic method genetic algorithm (GA).

The results intended to be achieved are;

- Classification of spyware using network packets
- Selecting the most meaningful parameters from network packets using meta-heuristic a genetic algorithm
- Conducting preliminary studies on developing a new spyware module by examining existing spyware.
- Introducing a new benchmark data set to the literature

2. Material and Method

This section describes the experimental setup and how the network packets of spyware are collected with the help of the experimental setup. After collecting the network packets, it is shown how the network packet features are extracted and how NCA and GA feature extractors are applied to these features. Finally, it is shown how the obtained features are classified with Decision trees.

Data Collection

In this study, an experimental environment was established using Firat University Faculty of Technology Forensics Engineering Department Laboratories to detect spyware from network packets. The experimental setup set up to collect the data set is shown in Figure 4. As can be seen in Figure 4, spyware was installed on Windows devices in the laboratory environment. At the same time, this spyware will transfer personal data to the server over the network. During the operation of the system, network packets were collected using the Wireshark program. Our benchmark data set was created by collecting the 4-class data in Table 2 with the help of the experimental setup. In total, there are 3007 examples for 4 classes in our data set.

Table 2 Features of our created data set

Class Number	Class	Number of samples
0	Normal	800
1	Screenshot	800
2	Mouse Location	899
3	Keylogger	508

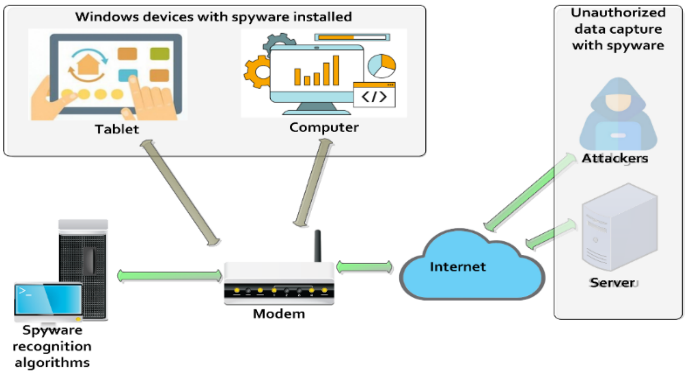

Windows devices with spyware installed
Tablet
Computer
Unauthorized data capture with spyware
Attackers
Server
Internet
Modem
Spyware recognition algorithms

Figure 4 Platform established to collect the data set and test the algorithms to be developed

Methodology

Within the scope of the study, it was aimed to use spyware that is widely used in the literature. For the first stage, it is planned to use spyware such as Spyera and Browser Hijacker.

Spyera tool

This software can perform many actions such as listening to the keyboard on the installed computer, taking application and desktop screenshots, and monitoring network connections [18].

Browser hijacker tool

It is malware that changes the settings, appearance, and behavior of the computer without the user's knowledge. It also makes it easier to perform activities such as data collection and keyboard reading. In the study, it was planned to develop a new spyware application by examining the structure and features of ready-made software. Existing and developed spyware programs were used to collect the data set. Network packages belonging to healthy and spyware types were collected by creating different scenarios. The Wireshark program was used to collect network packets [19].

Wireshark tool

It allows us to record and analyze the packets reaching computers over the network. It is a program that analyzes all TCP/IP messages on the Ethernet or modem layers connected to the computer [17], [20]. Features;

- Searches and filters packages based on various criteria.
- Shows packets with detailed protocol information.
- Helps you decrypt some protocols.

Network packets collected with the Wireshark program (.pcap files) must be parsed to be used. By parsing network packets, it is aimed to separate the data and process it easily. It is planned to use the Tshark program to parse the collected network packets [21].

Tshark tool

Tshark is a well-known and powerful command line tool and is used as a network analyzer. Developed by Wireshark. Its working structure is quite similar to tcpdump, unlike it it has some powerful decoders and filters. TShark can capture data packet information of different network layers and display it in different formats. TShark is used to analyze real-time network traffic and can read ".pcap" files to analyze information, examine the details of these connections, and help security professionals identify network problems [22].

The flow chart of the method developed for the detection and classification of spyware on the collected data set is given in Figure 5. Some spyware such as Spyera and Browser hijacker were installed on the computer forensics laboratory computer to be used as the test environment. By running this software, data was transferred to another computer via e-mail. While doing this process, ".pcap" files were collected with the Wireshark program, and their properties were obtained from the ".pcap" files through the Tshark program. The features obtained with the Tshark program were determined by examining studies in the literature. The features used by Tekin to detect network attacks [23]. The attributes used by Erginay for anomaly detection in network traffic [15].

Features used to detect malware in the literature have been researched and presented in tables. It is aimed to obtain and use these features on the data sets collected within the scope of the study. If there are many selected features, feature selection algorithms are used to select the most weighted features.

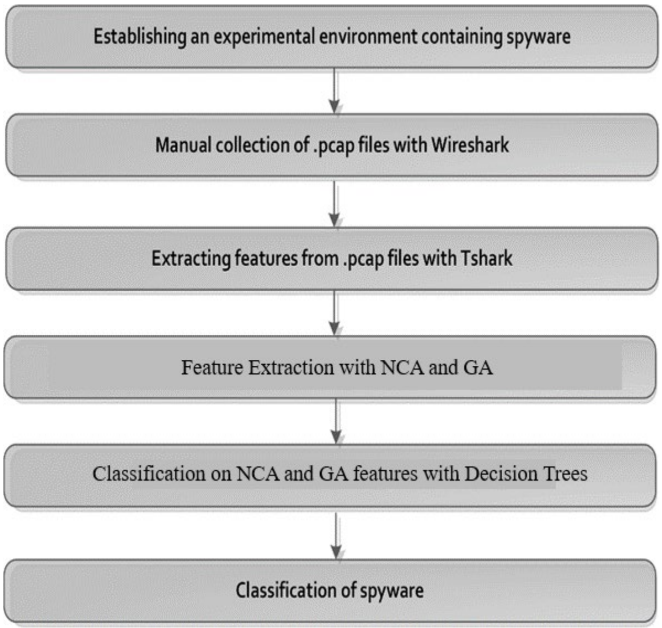

Establishing an experimental environment containing spyware
Manual collection of .pcap files with Wireshark
Extracting features from .pcap files with Tshark
Feature Extraction with NCA and GA
Classification on NCA and GA features with Decision Trees
Classification of spyware

Figure 5 Flow chart of the method developed within the scope of the study

As a result of research in the literature, it is seen that Chi^{2} [24], [25], NCA (Neighborhood Component Analysis) [26], [27], and ReliеF [28] algorithms are widely used.

- Chi^{2} is a statistics-based feature selection method. Chi^{2} test (X^{2}) calculates whether the relationship between two variables is dependent or independent. This method generally consists of two steps. In the first step, Chi2 statistics of features are calculated according to classes. In the second step, the degrees of freedom are calculated. By looking at the Chi^{2} values according to the determined threshold value, the features are separated until inconsistent features are found [25].
- NCA is a distance-based feature selection method [26], [27]. The NCA algorithm creates positive weights for each feature. Thus, it is different from other feature selection algorithms.
- Relief algorithm [28]. Kira et al. This algorithm, developed by [29], is widely used in the literature. This algorithm gives successful results for a two-class data set. Kononenko developed the ReliefF algorithm for multi-class datasets [30].

Deep learning and machine learning-based methods were used to detect and classify spyware by using the most weighted features obtained during the feature selection phase. The obtained performance results were compared with the literature. In our study, the features given in Table 5 were selected as features [31], [32].

Table 3 List of features selected for our study

No.	Feature name	Explanation
1	ip.version	IP address version
2	ip.hdr _len	IP address header length
3	ip.flags .rb	IP packet reserved bit
4	ip.flags .df	Unfragmented IP packet
5	ip.flags .mf	Multiple fragmented IP packets
6	ip.frag _offset	IP packet fragment offset value
7	ip.ttl	IP lifetime
8	ip.proto	IP protocol
9	ip.len	IP length
10	tcp.srcport	TCP source port
11	tcp.dstport	TCP destination port
12	tcp.seq	TCP sequence number
13	tcp.ack	TCP acknowledgment (ACK) number
14	tcp.len	TCP segment length
15	tcp.hdr _len	TCP header length
16	tcp.flags .fin	TCP Finnish flag
17	tcp.flags .syn	TCP Syn flag
18	tcp.flags .reset	TCP Reset flag
19	tcp.flags .push	TCP Push flag
20	tcp.flags .ack	TCP acknowledgment (ACK) flag
21	tcp.flags .urg	TCP Urgent flag
22	tcp.flags .cwr	TCP Reduced Congestion Window (CWR)
23	tcp.window _size	Calculated window size
24	tcp.urgent _pointer	TCP Urgent pointer
25	tcp.options .mss_val	TCP MSS value
26	frame it	Frame length

Our data set was classified using machine learning-based Decision Trees and Genetic Algorithms using the features given in Table 3. The Neighborhood Component Analysis (NCA) algorithm was used together with Decision Trees to evaluate its performance compared to the Genetic Algorithm (our recommended method).

Decision Trees (DT)

Decision Trees are a non-parametric trained machine learning method used for classification and regression. The goal is to create a model that predicts the value of the target variable by learning simple decision rules extracted from data set features. For example, decision trees learn from data to approximate a particular curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the more suitable the model gives results. Decision trees have a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes [33], [34]. The following are the working parameters for decision trees in this study:

- SplitCriterion: gdi,
- MaxNumSplit: 100,
- Surrogate: Off,
- ClassNames: [0,1,2,3]

Neighborhood Component Analysis (NCA)

Neighborhood component analysis is a trained machine learning method to classify multivariate data into different classes based on a specific distance measure on the data. Functionally, it serves the same purposes as the K-nearest neighbors algorithm and enables the direct use of so-called stochastic nearest neighbors. KBA is a distance metric learning method that uses the linear transformation of input data to maximize leave-one-out classification performance. The aim is to optimize the performance on future test data, but since the actual data distribution is unknown, leave-one-out = LOO performance is instead tried to be optimized in the training data [27], [35].

Genetic Algorithm (GA)

Genetic Algorithm is a meta-heuristic search and optimization algorithm in computer science and operations research that mimics the evolutionary processes seen in nature. GA starts with a population consisting of a set of individuals, a set of bits or characters that represent the solution to a problem. These individuals are evaluated using an objective function. The objective function measures how good a solution an individual is [36], [37].

The basic function of the genetic algorithm is to modify individuals in the population to produce better solutions. This is done through three main processes:

- **Selection:** The most fit individuals are selected to form a new population.
- **Crossover:** Two individuals are randomly selected and their genetic material is crossed. This can create new and better combinations.
- **Mutation:** An individual is randomly changed. This can create new and more creative solutions.

Figure 6 shows how selection is made with the Genetic Algorithm from candidate parents obtained from 26 different selected characteristics.

Figure 6 Use of Genetic Algorithm according to selected features

The following are the algorithm steps and working parameters for the genetic algorithm in this study:

- Population onset: 30
- Fitness function: select feedback accuracy
- Selection from the population: select according to high accuracy
- Crossover: crossover current population with selected population
- Mutation: mutation on crossover population

3. Experimental Results Discussion

In this study, classification was made using the Decision Trees algorithm with the features selected with the help of the Genetic Algorithm. Additionally, to make comparisons, classification was made using the Decision Trees algorithm using Neighborhood Components Analysis. As seen in Figure 7, the most weighted features were selected using the genetic algorithm for 200 iterations. In order to choose the highest accuracy, 10, 20, 30, 40, and 50 populations were used and accuracy values of 0.8909, 0.8923, 0.8929, 0.8919, and 0.8923 were obtained, respectively. When different population results were examined, it was seen that the most appropriate population number was 30. It is seen that selecting the most appropriate 18 features out of 26 features in our data set using the Genetic Algorithm gives better results.

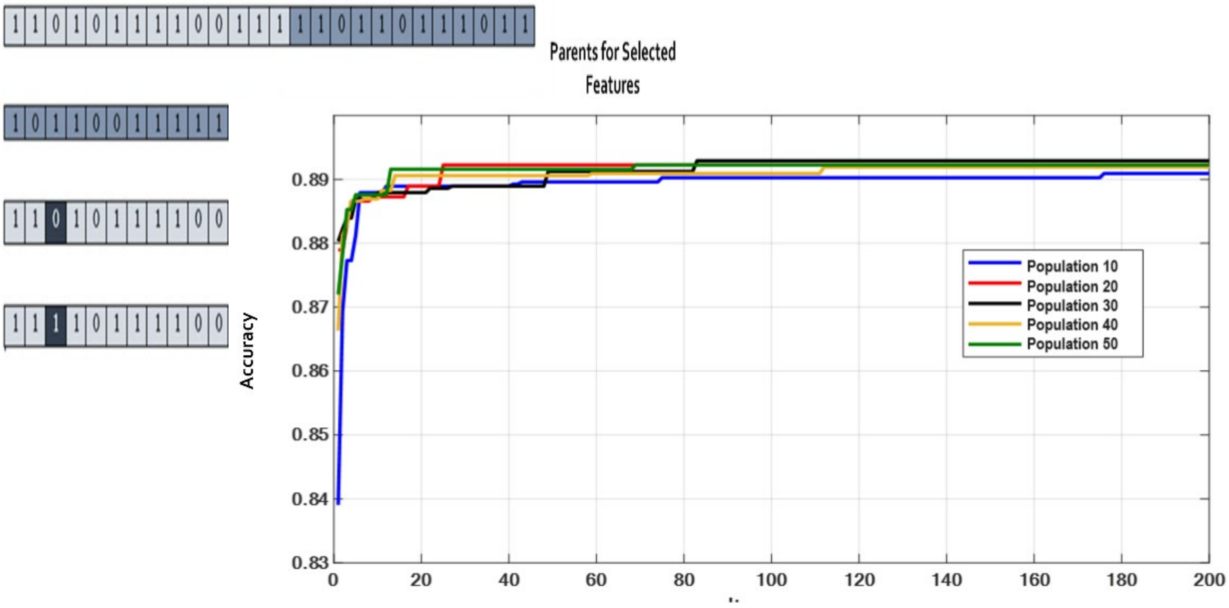


Figure 7 Accuracy plots over 200 iterations for different populations

18 features selected from 26 features in Table 3 by Genetic Algorithm; ip.version (1), ip.hdr_len (2), ip.flags.rb (3), ip.flags.df (4), ip.flags.mf (5), ip.frag_offset (6), ip. ttl (7), tcp.srcport (10), tcp.dstport (11), tcp.seq (12), tcp.len (14), tcp.hdr_len (15), tcp.flags.fin (16), tcp .flags.reset (18), tcp.flags.urg (21), tcp.window_size (23), tcp.urgent_pointer (24), framelen (26).

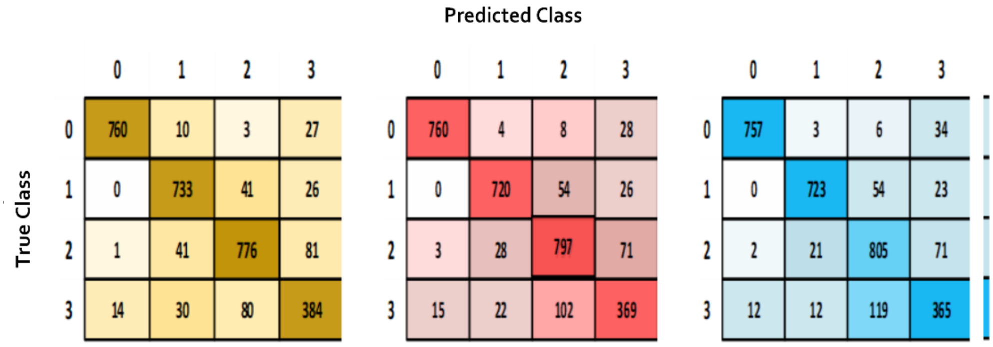
In Figure 8, confusion matrices are given for the class predictions of Decision Trees (DT), Neighborhood Components Analysis (NCA) + Decision Trees (KA), and Genetic Algorithm (GA) + Decision Trees (DT) methods, respectively. When the confusion matrices are examined, it is seen that much better results are obtained for classes 0, 1, 2 (Normal, Screen Display, and Mouse Position) in our data set than for class 3 (Keylogger).

Class-by-class accuracy values are given in Table 4. Table 5 shows the accuracy, precision, recall, F1-score, and geometric values obtained for all three methods in 100 iterations (DT, NCA + DT, GA + DT). Average (Geometric mean) performance metrics are given.

Table 4 Class-by-class accuracy values

Class	DT (%)	NCA + DT (%)	GA + DT (%)
0	95.0	95.0	94.62
1	91.62	90.0	90.37
2	86.31	88.65	89.54
3	75.59	72.63	71.85

When Table 5 is examined, it is seen that the method we recommend (GA + DT) gives the best result (89.35%, 89.07%, 88.39) in terms of accuracy, precision, and F1-score values. The best results (87.95%, 87.72%) in terms of recall and geometric mean values were obtained with the NCA + DT method.



	0	1	2	3
0	760	10	3	27
1	0	733	41	26
2	1	41	776	81
3	14	30	80	384

a)

	0	1	2	3
0	760	4	8	28
1	0	720	54	26
2	3	28	797	71
3	15	22	102	369

b)

	0	1	2	3
0	757	3	6	34
1	0	723	54	23
2	2	21	805	71
3	12	12	119	365

c)

Figure 8 Confusion matrices a) DT b) NCA + DT c) GA + DT

Table 5 Performance values of our method and other methods for 100 iterations

Methods	Metrics	Accuracy	Precision	Recall	Geometric Mean	F1-Score
DT	Max	88.22	87.11	87.13	86.81	87.12
	Min	87.99	86.95	86.83	86.49	86.89
	Mean	88.11	87.03	86.98	86.64	87.01
	Std	0.11	0.08	0.14	0.159	0.11
NCA + DT	Max	88.92	88.22	**87.95**	**87.72**	87.98
	Min	86.86	85.75	85.25	84.66	85.56
	Mean	88.00	86.98	86.79	86.44	86.88
	Std	0.32	0.37	0.38	0.43	0.368
Proposed Method (GA + DT)	Max	**89.35**	**89.07**	87.72	87.24	**88.39**
	Min	86.99	86.08	85.31	84.63	85.69
	Mean	88.10	87.37	86.46	85.91	86.91
	Std	0.33	0.42	0.35	0.41	0.37

Descriptions of the performance metrics in Table 5 and calculation equations (1-5) are given below. Accuracy, precision, recall, geometric mean, and F-measure/score were selected to calculate performance rates comprehensively. These performance metrics were calculated by using the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Mathematical notations of the used performance metrics were shown in Eqs. 1-5.

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

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

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

$$Geometric\_Mean = \sqrt{\frac{TP*TN}{(TP+FN)*(TN+FP)}} \tag{4}$$

$$F1-Score = \frac{2TP}{2TP+FP+FN} \tag{5}$$

When the confusion matrices given in Figure 8 are examined, it can be seen that the Normal and Screenshot classes are classified very well. The classification for Keylogger needs to be improved.

4. Conclusion

Considering the confusion matrix results in Figure 8, our data set needs to be increased for the 3rd class (type) in our data set (keylogger). When Table 7 is examined, it is seen that the method we recommend (GA + DT) gives better results compared to other methods. The contribution of the Genetic Algorithm (GA) here has been shown to provide better performance in terms of time by reducing the number of features. It is seen that the NCA + DT method gives very close results to the method we recommend. Despite the limited benchmark data set size and lack of balance, the results are satisfactory.

In future studies, it will be investigated how the results will be for larger populations other than the 10,20,30,40,50 populations. Furthermore, the potential influence of alternative meta-heuristic methodologies on this issue will be examined. Additionally, it should be investigated what results our proposed method will yield for higher iterations. It is thought that better results will be obtained by using deep learning methods with our expanded data set. In future studies, it is aimed to develop software that detects whether there is spyware or not, and if so, what type of spyware it is.

Acknowledgement: This study was supported by TUBITAK 2209/A. Project number: 1919B012219445.

Competing interests

The authors declare that they have no competing interests.

References

[1] G. Canbek and Ş. Sağıroğlu, “Kötücül ve Casus Yazılımlar: Kapsamlı bir Araştırma,” *J. Fac. Eng. Archit. Gazi Univ.*, vol. 22, no. 1, pp. 121–136, 2007.

[2] K. Pandey, M. Naik, J. Qamar, and M. Patil, “Spyware Detection Using Data Mining,” *Int. J. Eng. Tech.*, vol. 1, no. 2, pp. 5–8, 2015.

[3] S. Yadav and P. R. Randale, “Detection and Prevention of Keylogger Spyware Attack,” *Int. J. Adv. Found. Res. Sci. Eng.*, vol. 1, pp. 1–5, 2015.

[4] İ. Bulut, “Analiz Sürecini Atlatmaya Çalışan Zararlı YAzılımlar ve Derin Öğrenme Temelli Zararlı Yazılım Tespiti,” Yıldız Teknik Üniversitesi, 2017.

[5] C. A. Dinçer and İ. A. Doğru, “Android Kötücül Yazılım Tespiti Yaklaşımları,” *Uluslararası Bilgi Güvenliği Mühendisliği Derg.*, no. 2, pp. 48–58, 2017.

[6] A. Utku, “Using network traffic analysis deep learning based Android malware detection,” *J. Fac. Eng. Archit. Gazi Univ.*, vol. 37, no. 4, pp. 1823–1838, 2022, doi: 10.17341/gazimmfd.937374

[7] A. Mehtab *et al.*, “AdDroid: Rule-Based Machine Learning Framework for Android Malware Analysis,” *Mob. Networks Appl.*, vol. 25, no. 1, pp. 180–192, 2020, doi: 10.1007/s11036-019-01248-0

[8] A. Pektaş and T. Acarman, “Deep learning for effective Android malware detection using API call graph embeddings,” *Soft Comput.*, vol. 24, no. 2, pp. 1027–1043, 2020, doi: 10.1007/s00500-019-03940-5

[9] K. Bakour and H. M. Ünver, “DeepVisDroid: android malware detection by hybridizing image-based features with deep learning techniques,” *Neural Comput. Appl.*, vol. 33, no. 18, pp. 11499–11516, 2021, doi: 10.1007/s00521-021-05816-y

[10] M. Tokmak and E. U. Küçüksille, “Detection of Windows Executable Malware Files with Deep Learning,” *Bilge Int. J. Sci. Technol. Res.*, vol. 3, pp. 67–76, 2019, doi: 10.30516/bilgesci.531801

[11] C. K. Bauri, C. Indulkar, S. Jadhav, and P. A. S. Khandagale, “A Survey on Windows Post Exploitation [MSF] Keylogger for Security,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 3, pp. 721–726, 2022, doi: 10.22214/ijraset.2022.40684

[12] D. Javaheri, M. Hosseinzadeh, and A. M. Rahmani, “Detection and elimination of spyware and ransomware by intercepting kernel-level system routines,” *IEEE Access*, vol. 6, pp. 78321–78332, 2018, doi: 10.1109/ACCESS.2018.2884964

[13] M. . NarasimaMallikarajunan.K., Preethi.S.R, Selvalakshmi.S, and Nithish.N, “Detection of Spyware in Software Using Virtual Environment,” in *Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019)*, 2019, pp. 1138–1142.

[14] M. Dama, “Windows Fonksiyonları Kullanılarak Özgün Bir Casus Yazılım Tasarımı ve Alınabilecek Önlemler,” Gazi Üniversitesi, 2014.

[15] E. Erginay, “Ağ trafiğinde anormallik tespiti için veri seti oluşturma ve test yöntemlerinin karşılaştırılması,” Gazi Üniversitesi, 2019.

[16] P. McLaren, G. Russell, and B. Buchanan, “Mining malware command and control traces,” *Proc. Comput. Conf. 2017*, vol. 2018-Janua, no. July, pp. 788–794, 2018, doi: 10.1109/SAI.2017.8252185

[17] W. Ames, “Understanding Spyware : Risk and Response,” *Security*, no. October, pp. 1–12, 2005.

[18] “Spyera,” 2023. Available: https://spyera.com/tr/. [Accessed: Nov. 01, 2023]

[19] “Browser Hijacker.” Available: https://www.malwarebytes.com/blog/threats/browser-hijacker. [Accessed: Nov. 01, 2023]

[20] S. Wang, “Analysis and Application of Wireshark in TCP/IP Protocol Teaching,” *2010 Int. Conf. E-Health Netw. Digit. Ecosyst. Technol.*, vol. 2, pp. 269–272, 2010.

[21] U. Lamping, R. Sharpe, and E. Warnicke, “Wireshark User’s Guide,” 2004.

[22] “Turkhackteam,” 2023. Available: https://www.turkhackteam.org/forumlar/siber-guvenlik.538/. [Accessed: Dec. 01, 2023]

[23] R. Tekin, “Nesnelerin İnterneti Uygulamaları için

Saldırı Tespit Yöntemlerinin Geliştirilmesi," Fırat Üniversitesi, 2022.

[24] H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proceedings of the International Conference on Tools with Artificial Intelligence*, 1995. doi: 10.1109/tai.1995.479783

[25] B. Yazıcı, F. Yaslı, H. Y. Gürleyik, and U. O. Turgut, "Veri Madenciliğinde Özellik Seçim Tekniklerinin Bankacılık Verisine Uygulanması Üzerine Araştırma ve Karşılaştırmalı Uygulama," pp. 72–83, 2015.

[26] T. Tuncer and F. Ertam, "Neighborhood component analysis and reliefF based survival recognition methods for Hepatocellular carcinoma," *Phys. A Stat. Mech. its Appl.*, vol. 540, p. 123143, 2020, doi: 10.1016/j.physa.2019.123143

[27] O. Yaman, "An automated faults classification method based on binary pattern and neighborhood component analysis using induction motor," *Meas. J. Int. Meas. Confed.*, 2021, doi: 10.1016/j.measurement.2020.108323

[28] T. Tuncer, S. Dogan, and F. Ozyurt, "An automated Residual Exemplar Local Binary Pattern and iterative ReliefF based COVID-19 detection method using chest X-ray image," *Chemom. Intell. Lab. Syst.*, no. January, 2020.

[29] K. Kira and L. A. Rendell, "Feature selection problem: traditional methods and a new algorithm," in *Proceedings Tenth National Conference on Artificial Intelligence*, 1992, pp. 129–134.

[30] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1994. doi: 10.1007/3-540-57868-4_57

[31] "Display Filter Reference: Internet Protocol Version 4." Available: https://www.wireshark.org/docs/dfref/i/ip.html. [Accessed: Nov. 15, 2023]

[32] "Display Filter Reference: Transmission Control Protocol." Available: https://www.wireshark.org/docs/dfref/t/tcp.html. [Accessed: Nov. 15, 2023]

[33] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, 1986, doi: 10.1023/A:1022643204877

[34] B. Kamiński, M. Jakubczyk, and P. Szufel, "A framework for sensitivity analysis of decision trees," *Cent. Eur. J. Oper. Res.*, 2018, doi: 10.1007/s10100-017-0479-6

[35] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*, 2005.

[36] M. Melanie, "An introduction to genetic algorithms By Melanie Mitchell. MIT Press, Cambridge, MA. (1996). 205 pages. $30.00," *Comput. Math. with Appl.*, 1996, doi: 10.1016/S0898-1221(96)90227-8

[37] G. D.E., "Genetic algorithms in search, optimization, and machine learning," *Mach. Learn. Reading, Mass, Addison-Wesley Pub. Co*, 1998.