http://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Series A2-A3
Volume 66, Number 1, Pages 26-44 (2024)
DOI: 10.33769/aupse.1320795
ISSN 1303-6009 E-ISSN 2618-6462

COMMUNICATIONS
SERIES A2-A3

Research Article; Received: June 28, 2023; Accepted: August 31, 2023

USB-IDS-1 DATASET FEATURE REDUCTION WITH GENETIC ALGORITHM

Mustafa Veysel ÖZSARI[1], Şifa ÖZSARI[1], Ayhan AYDIN[1],
Mehmet Serdar GÜZEL[1]

[1]Ankara University, Computer Engineering Department, Ankara, 06830, TÜRKİYE

Abstract. Technology and online opportunities brought by technology are increasing day by day. Many transactions, from banking to shopping, can be done online. However, the abuse of technology is also increasing at the same rate. Therefore, it is very important to ensure the security of the network for data protection. The application of artificial intelligence-based approaches has also become popular in the field of information security. When the data collected for intrusion detection is examined, it is seen that there are many features. In this study, the features in the USB-IDS-1 dataset were reduced by genetic algorithm and its success was examined with various classifiers. Among the selected methods, there are decision trees, random forest, k-NN, Naive Bayes and artificial neural networks. Accuracy, sensitivity, precision and F1-score were used as metrics. According to the results obtained, it was seen that the genetic algorithm was quite successful in the Hulk and Slowloris data set, it was partially effective in the Slowhttptest data, but was not successful in the TCP set. However, the performance of the algorithms was poor as a result of using all features in Slowhttptest and TCP data.

1. Introduction

In the last 10 years, technology has progressed very rapidly. With this progress, many processes from health to defense, from online shopping to engineering applications have started to be done online. This has led to a proportional increase in the number of data entered. However, it has also increased in malicious attacks. Since great number of personal information is stored online, it is crucial to protect this data. One of the popular tools for information security is Intrusion Detection

Keywords. Intrusion detection, USB-IDS-1, genetic algorithm, machine learning.

✉ ozsaris@ankara.edu.tr-Corresponding author; 0000-0002-0531-4645
✉ veysel.ozsari.95@gmail.com; 0000-0002-1997-9971
✉ ayaydin@ankara.edu.tr; 0000-0001-7938-0509
✉ mguzel@ankara.edu.tr; 0000-0002-3408-0083.

Systems (IDS). IDS, in general terms, are devices or software that detect malicious attacks on systems.

As in many fields, studies are carried out on Artificial Intelligence (AI) and especially Machine Learning (ML) based approaches. AI was developed with inspiration from the learning process of human brain. Kaplan et al. [1] defined AI as the ability of a system to accurately interpret data, learn from such data, and use this information to achieve specific goals and tasks. ML is a sub-field of artificial intelligence and was first introduced by Arthur Samuel [2]. Today's ML applications are generally about developing a classifier with using the available data and then with these models to produce predictions for new (unseen) input. When the literature is examined, ML-based studies [3-7] in intrusion detection and KDD Cup99 [8], CAIDA [9], NSL-KDD [10] data sets can be given as examples for data prepared for IDS.

In presented paper, we conducted a study on the USB-IDS-1 [11] dataset introduced in 2021. Section 2 provides detailed information about this data set. It is a big data and has 83 attributes (features). The main problem encountered in processing such large-sized data is hardware inadequacy and time. If the features of the devices used are powerful, faster results can be produced. However, this is not always possible. In this study, we performed feature selection on the USB-IDS-1 dataset using Genetic Algorithm (GA) [12] and then carried out classification with these features.

The genetic algorithm developed by Holland is an optimization algorithm. The main object in optimization approaches is to select the best solution among all alternatives for a specific task. GAs is a widely used and simple optimization method in this field. Therefore, this algorithm was utilized for feature reduction. Thus, the features that have no effect on the classification were identified and eliminated, and a classification system was designed to predict the new data without using these features.

The rest of the article is organized as follows: In Section 2, detailed information about the dataset and algorithms used in the study is given. In Section 3, parameter settings, experiments and test results are explained. Section 4 is the results and the article is concluded.

2. Material and Methods

2.1. **Dataset.** There are various data sets that are widely used for academic studies and examinations on STS in the literature. In this research, we used the USB-IDS-1 dataset introduced by Catillo et al. [11] in 2021. This dataset consists of 83 features and 16 classes. These features are: “Flow ID”, “Src IP”, “Src Port”, “Dst IP”, “Dst Port”, “Protocol”, “Timestamp”, “Flow Duration”, “Total Fwd Packet”, “Total Bwd packets”, “Total Length of Fwd Packet”, “Total Length of Bwd Packet”, “Fwd Packet Length Max”, “Fwd Packet Length Min”, “Fwd Packet Length Mean”, “Fwd Packet Length Std”, “Bwd Packet Length Max”, “Bwd Packet Length Min”, “Bwd Packet Length Mean”, “Bwd Packet Length Std”, “Flow Bytes/s”, “Flow Packets/s”, “Flow IAT Mean”, “Flow IAT Std”, “Flow IAT Max”, “Flow IAT Min”, “Fwd IAT Total”, “Fwd IAT Mean”, “Fwd IAT Std”, “Fwd IAT Max”, “Fwd IAT Min”, “Bwd IAT Total”, “Bwd IAT Mean”, “Bwd IAT Std”, “Bwd IAT Max”, “Bwd IAT Min”, “Fwd PSH Flags”, “Bwd PSH Flags”, “Fwd URG Flags”, “Bwd URG Flags”, “Fwd Header Length”, “Bwd Header Length”, “Fwd Packets/s”, “Bwd Packets/s”, “Packet Length Min”, “Packet Length Max”, “Packet Length Mean”, “Packet Length Std”, “Packet Length Variance”, “FIN Flag Count”, “SYN Flag Count”, “RST Flag Count”, “PSH Flag Count”, “ACK Flag Count”, “URG Flag Count”, “CWR Flag Count”, “ECE Flag Count”, “Down/Up Ratio”, “Average Packet Size”, “Fwd Segment Size Avg”, “Bwd Segment Size Avg”, “Fwd Bytes/Bulk Avg”, “Fwd Packet/Bulk Avg”, “Fwd Bulk Rate Avg”, “Bwd Bytes/Bulk Avg”, “Bwd Packet/Bulk Avg”, “Bwd Bulk Rate Avg”, “Subflow Fwd Packets”, “Subflow Fwd Bytes”, “Subflow Bwd Packets”, “Subflow Bwd Bytes”, “FWD Init Win Bytes”, “Bwd Init Win Bytes”, “Fwd Act Data Pkts”, “Fwd Seg Size Min”, “Active Mean”, “Active Std”, “Active Max”, “Active Min”, “Idle Mean”, “Idle Std”, “Idle Max”, “Idle Min”, “Label”.

The attributes “Flow ID”, “Fwd Header Length”, “Src IP”, “Src Port”, “Dst IP”, “Dst Port”, “Timestamp” are metadata and are not used for classification. These columns were removed from the dataset so that they do not affect the classification and feature selection result. In addition, rows with NaN values were excluded from the data set. Class labels also include the defense module, and the groups for defense are as follows:

1. Hulk-NoDefense
2. Hulk-Reqtimeout
3. Hulk-Evasive
4. Hulk-Security2
5. TCPFlood-NoDefense
6. TCPFlood-Reqtimeout
7. TCPFlood-Evasive
8. TCPFlood-Security2
9. Slowhttptest-NoDefense
10. Slowhttptest -Reqtimeout
11. Slowhttptest -Evasive
12. Slowhttptest -Security2
13. Slowloris-NoDefense

14. Slowloris-Reqtimeout
15. Slowloris-Evasive
16. Slowloris-Security2

In here, the part before – denotes the attack type, and the next part denotes the defense model. Attack and defense types are briefly explained as follows [11]:

- *Hulk: This type of attack generates a large number of unique requests. Thereby, it is intended to prevent the server from recognizing a pattern and filtering the attack. This makes it difficult to detect requests from the signature.*
- *TCPFlood: TCPFlood is a well-known DoS attack tool that is considered as a flood attack. In here, the attacker's requests lock the available ports on the server and cause TCP connections from legitimate clients to not be accepted.*
- *Slowhttptest: The tool used in this attack type allows the launch of slow DoS application layer attacks. HTTP connections can be extended in different ways. In the experiments for this dataset, Slowhttptest was used in "slowloris" mode, which sends incomplete HTTP requests to the target server.*
- *Slowloris: In this attack type, DoS attacks are produced by sending slow HTTP requests to the server. It uses low-bandwidth approaches that exploit a weakness in the TCP fragmentation management of the HTTP protocol.*
- *Reqtimeout: This mod_reqtimeout defense module is intended to protect the HTTP server from slow DoS attacks like Slowloris. This module allows determining the minimum data rate and timeouts required to keep a connection open.*
- *Evasive: This mod_evasive module has been developed to protect the server from attacks (such as Hulk) that try to make the server unusable by consuming the server's resources with a large number of requests. It monitors incoming requests and looks for suspicious IPs and similar activities. For example, events such as multiple requests within a short period of time or multiple requests per second for the same pages. If any of these events are detected, a 403-warning code will be responded to and the IP will be blacklisted for a certain period of time [13].*
- *Security2: The mod_security2 module, which is based on a set of rules related to known attack structures that can be obtained from free or pay-as-you-go repositories, acts as a kind of intrusion detection and prevention system.*

In the presented study, Hulk, TCP, Slowhttptest and Slowloris attack types were taken as a separate group. Because there is not an equal number of rows from all groups in the data set. Figure 1 shows the distribution of the categories.

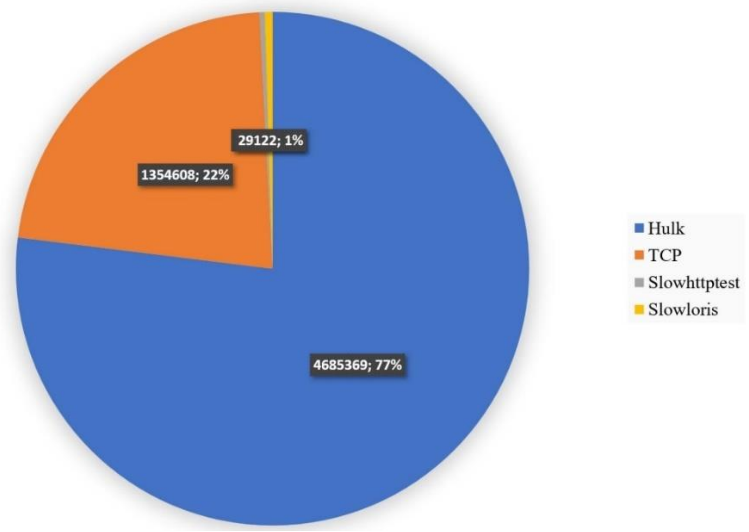


FIGURE 1. Data distribution.

In the literature, researches on USB-IDS-1 are quite few. First of the studies we examined is the [14]. In this paper, experiments were carried out by applying decision trees, random forest and deep neural network algorithms on the USB-IDS-1. Catillo et. al. performed the training process on a different dataset. They carried out the test on presented data set.

Another important study is [15]. A new approach for IoT security monitoring that combines deep autoencoder models with Explainable Artificial Intelligence (XAI) is presented by Kalutharage et al. It is aimed to verify the reliability and robustness of ML-based intrusion detection systems. The proposed method was tested using the USB-IDS-1 dataset.

2.2. **Machine Learning.** Today, machine learning, which has been used in many areas from health to social media, from banks to online shopping, is a system that develops itself according to the data it has acquired. ML algorithms are divided into 3 categories according to types [16]:

- *Supervised learning: They are ML algorithms in which learning is made by using the output data corresponding to the input data.*
- *Unsupervised learning: They are algorithms that process with using only input data. No output data is given to the systems operating in this way.*
- *Reinforced learning: In systems working with this method, the system is rewarded depending on its performance. According to this award, the algorithm updates itself.*

In this study, decision trees, random forest [17], k-Nearest Neighbor (KNN) [18], Navie Bayes and Artificial Neural Networks (ANN) algorithms were employed. All of them are supervised learning approaches.

Decision trees are a tree-like ML algorithm, which basically consists of decision nodes and leaf nodes. They are widely used for classification and regression tasks. In decision trees, data is divided into sub-parts in the form of nodes. The first of these division is the root node. It proceeds from the root node to the child nodes according to the "Yes" or "No" status. Likewise, in child nodes, according to "Yes" or "No", it moves forward to the next node until the leaf node is reached. Leaf nodes are nodes with classes (indicating the class of the data). Decision trees are widely used in IDS studies because they are applicable in complex data sets and have produced successful results [19].

Random forests (random decision forests) are a tree-based ML algorithm developed by Breiman [17]. They are widely used for problems such as classification or regression. A random forest makes classification or regression by creating a large number of decision trees during the training phase. A decision tree is created for each class and classification is made by considering the results of these sub-decision trees. An example for forest structure is shown in Figure 2.

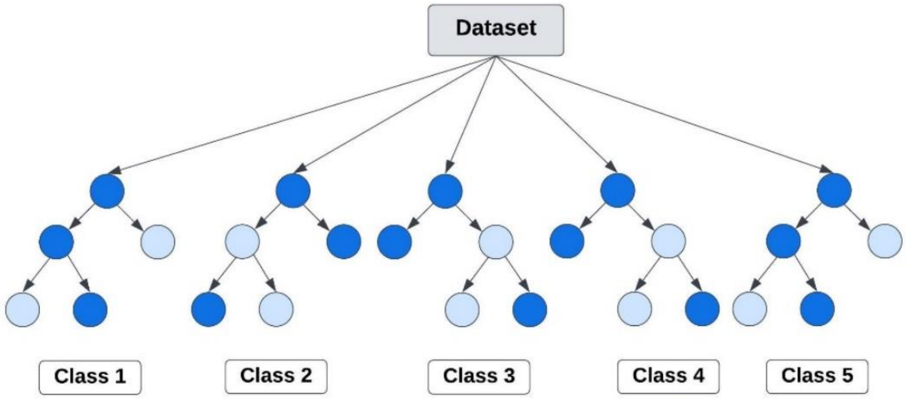


FIGURE 2. Random forest.

The KNN [18] algorithm is an algorithm that classifies a new input data according to its proximity to previous data. The number k represents the number of nearest neighbors to be checked. For example, if k is taken as 10, the class of the new incoming data is determined by looking at the class of 10 nearest neighbors. Here, the k parameter is important and directly affects the result. Various formulas such as Minkowski, Manhattan, Euclidean etc. are used when calculating proximity to neighbors.

Navie Bayes is an ML algorithm that runs based on the conditional probability calculation formula introduced by Thomas Bayes in 1812. This approach, which can also be applied to unbalanced data sets, basically makes a classification by calculating probability for each element. It is tried to determine the category of the new test samples given to the system with the probability operations performed on the data used to train the system. As with other ML-based approaches, the more data in this method, the more reliable the results.

Artificial neural networks, inspired by the neurons in the human brain, are a network of interconnected artificial neurons. Each neuron is connected to each other by weights and information is stored in these weights. A neuron produces output by processing the input it receives in various ways. This output is either the input of another neuron or the output of the net. A basic neural network structure is presented in Figure 3. In here, data is presented to the network from the input layer, then transmitted to the hidden layer(s), finally passing through the output layer to produce an output. The number of neurons in the input and output layer is determined depending on the input and output data of the problem. The number of hidden layers in the network is not fixed, it is adjusted according to the problem. The high number of hidden layers and neurons in the hidden layer may increase accuracy. However, it also causes high computational costs.

2.3. **Genetic Algorithm.** Optimization is the search for the most suitable solution among more than one solution according to a specific purpose. GA [12], a popular optimization algorithm, was first proposed by Holland in 1975. This approach is based on the theory of evolution and consists of 3 basic steps: selection, crossover and mutation. According to the theory of evolution, dominant individuals are passed on to the next generation, while weak individuals perish. Figure 4 shows the pseudocode of the GA.

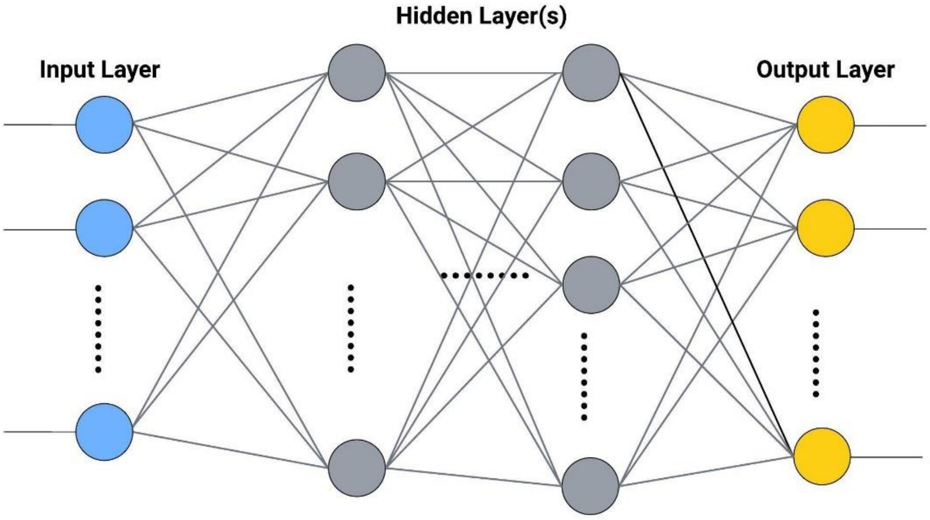
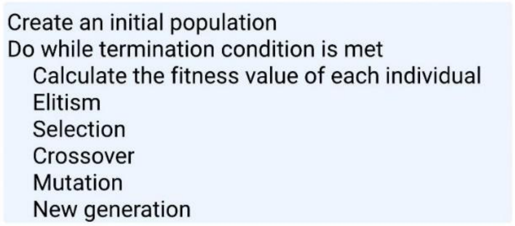


FIGURE 3. Basic ANN structure.



```
Create an initial population
Do while termination condition is met
    Calculate the fitness value of each individual
    Elitism
    Selection
    Crossover
    Mutation
    New generation
```

FIGURE 4. Pseudocode of the GA [20].

When we detail the pseudocode in Figure 4, first the population is randomly generated in accordance with the problem. Until the termination condition is met: The fitness value of individuals is computed. The fitness value is calculated using the objective function for problem. Individuals with the best fitness value (at a certain rate) are passed on to the next generation without any change. This process is called elitism and is not mandatory. Then, the individuals who will form the crossover pool are determined according to a specific selection method. Children are formed by using the crossover method determined between these individuals. The gene exchange process between two individuals is called crossover. Mutation process is applied to some of the offspring individuals. A mutation is a gene change in an individual and is generally applied at a low rate. The high rate of mutation can cause the loss of good individuals. Termination in GAs is done in various ways:

- When a certain number of iterations is reached, the run is terminated.

- The experiment is terminated when a targeted success rate is achieved.
- If the best fitness value does not change during a certain iteration, termination is made.

3. Results and Discussion

In this section, parameter settings, experiments and test results are given. All experiments were run in the Google Colaboratory (COLAB) [21] environment. COLAB is a product developed by Google Research where python codes can be written and run online. It allows automatic use of GPU and many libraries, so it is very practical for machine learning related studies [22].

3.1. **Metrics and Parameter Settings.** Accuracy rate, recall, precision and F1-score metrics were used to observe the performance of the methods. Formulas for these metrics are given in Equations 1, 2, 3 and 4, respectively

$$Accuracy\,rate = \frac{100*(TP+FP)}{TP+FP+TN+FN} \tag{1}$$

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F1-score = \frac{2*\text{recall}*\text{precision}}{\text{recall}+\text{precision}} \tag{4}$$

For Equations 1, 2 and 3, TP corresponds to True Positive (classify healthy test as sick), FP False Positive (classify healthy sample as sick), TN True Negative (class healthy sample as healthy), and FN corresponds to False Negative (categories sick as healthy). The parameters of the methods used in the study were determined by the preliminary experiments and were adjusted as follows:

- *GA: The population size was set to 300, the iteration number to be 100, and the mutation rate to be 0.1. Elitism was made by transferring the best 2 individuals to the next generation. Tournament selection method, two-point crossover and value-changing mutation methods were used.*
- *Decision tree: The root node was determined using the Gini formula.*
- *Random forest: The number of trees is taken as 100.*
- *k-NN: Nearest 100 neighbors were checked.*

- *ANN: ADAM [23] was used as the solver and "identity" was selected as the activation function. 4 hidden layers (100-500 nodes each) were added to the net.*

In the tournament selection technique, the individual with the highest fitness value is taken from two randomly selected individuals. The selection process is performed in this way until the parent individuals to be used in the crossover are completed. In a two-point crossover, the genes of two parents are exchanged between two points. Two random points are chosen and the genes between these two points are exchanged between the two parent individuals. In the value-changing mutation technique, the value of a randomly selected gene of the individual is changed. While selecting the individual to be mutated, a random number between 0 and 1 is first generated for this individual. If this number is greater than the mutation rate, this individual is mutated. If it is small, no mutation is applied to the individual.

3.2. **Experiments.** After parameter settings were completed, classification was performed without making any feature selection at first. In other words, algorithms made classification using all attributes. Then, the performance of GA in feature reduction was examined. Firstly, separate experiments were carried out for each attack type. The main reason for conducting experiments in this way is the number of data, and the data distribution is shown in Figure 1. As can be seen from Figure 1, the data numbers of the classes are quite unbalanced. Data distribution is very important in machine learning-based approaches. The unbalanced data can cause the algorithms to make wrong inferences. For example, the Hulk group is dominant and will reduce the impact of other categories in the classification. For this reason, data were taken equally for each class, taking into account its own count of data, and separate experiments were carried out. Then, in order to observe the performance of the approaches in a way that all groups were included, the same experiments were performed again by taking equal amount of data from each group based on the lowest data count. The results of only 10 features were presented for experimentation with all attributes. Both the not enough number of data and the quadrupling of the number of classes affected the results negatively for 5 features and very poor results were obtained. Therefore, in Table 5, the outputs for 5 attributes were not included, only the results for 10 features were mentioned.

In feature selection part, firstly, the number of features was reduced to 5. When poor results were obtained in 5, a performance of the algorithms was examined for 10. A population was formed in accordance with GA as shown in Figure 5. In here, sample individuals of reducing the count of features to 5 are shown. Therefore, an individual consists of 5 genes. Each gene is a number which correspond to an attribute.

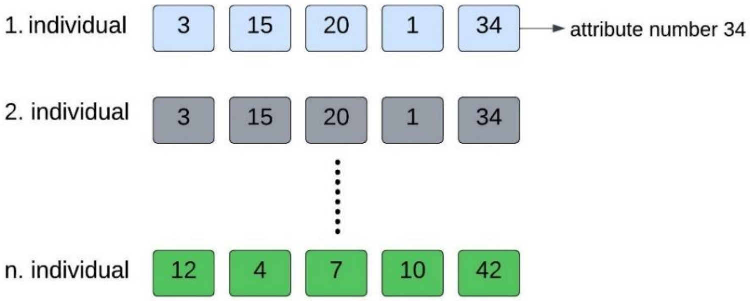


FIGURE 5. Population.

The fitness value is calculated for each individual, then crossover is applied to the individuals selected by the tournament selection method. An example crossover is shown in Figure 6. The genes (in the range indicated by lines) of the two parent individuals in Figure 6 (a) are replaced, and two new offspring individuals in (b) are obtained. Then the mutation stage is passed. A new population is obtained after the value-changing mutation process is completed. These operations (elitism, crossover, mutation) are performed sequentially up to 100 iterations. With the termination of the algorithm, the most successful individual is taken as the solution of the problem.

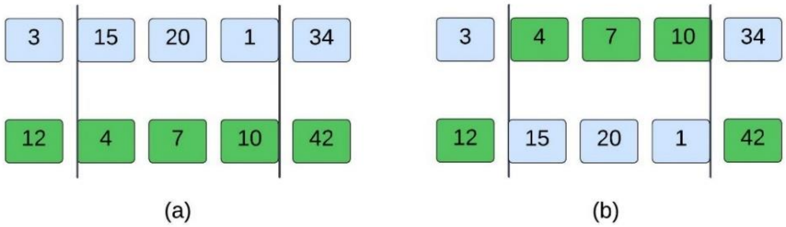


FIGURE 6. Crossover (a) parents (b) children.

After the completion of the experiments, the performance examination was carried out. In Tables 1, 2, 3 and 4, the results obtained with feature reduction and without feature reduction for Hulk, TCP, Slowhttptest, Slowloris are given respectively. In Table 5, the outputs of the experiments in which all groups were included are presented.

TABLE 1. Experiments for Hulk.

Features	**Classifier**	**Accuracy rate**	**Recall**	**Precision**	**F1-score**
5	Decision tree	0,71	0,72	0,72	0,72
All	Decision tree	0,72	0,73	0,73	0,73
5	Random forest	**0,72**	**0,73**	**0,73**	**0,73**
All	Random forest	**0,73**	**0,73**	**0,74**	**0,74**
5	k-NN	0,65	0,65	0,66	0,65
All	k-NN	0,66	0,66	0,68	0,66
5	Bayes	0,71	0,72	0,72	0,72
All	Bayes	0,72	0,73	0,73	0,73
5	ANN	0,62	0,62	0,64	0,61
All	ANN	0,64	0,63	0,65	0,63

When Table 1 is examined, it is seen that the random forest algorithm gives the best result in all metrics in terms of both taking all features and feature reduction. It is seen that GA is successful in feature reduction with results of 0.72 and above. 5 features in terms of their effect on classification accomplishment are:

- “Down/up ratio”
- “Fwd IAT Min”
- “Bwd IAT Mean”
- “Bwd IAT Max”
- “Bwd Packet Length Mean”

When other approaches are examined, it is seen that the decision tree and Bayesian are as successful as the random forest. Next comes k-NN and the least effective method is ANN. When all values in the table are examined, results of 61% and above are acceptable.

When Table 2 is analyzed, it is seen that the algorithms could not produce effective results in terms of both taking all features and feature reduction. Values of 0.25 and below are not acceptable. It is noteworthy that there is a decrease of up to 0.1 in the F1-score. All methods yielded more or less the same outputs. This situation is thought to be caused by the data set rather than the algorithms. However, it can be deduced from the table that the decision tree and random forest algorithm outperform other algorithms.

TABLE 2. Experiments for TCP.

Features	**Classifier**	**Accuracy rate**	**Recall**	**Precision**	**F1-score**
5	Decision tree	0,25	0,25	0,25	0,25
10	Decision tree	0,25	0,26	0,25	0,25
All	Decision tree	0,25	0,25	0,25	0,25
5	Random forest	0,26	0,26	0,26	0,26
10	Random forest	0,25	0,25	0,25	0,25
All	Random forest	0,25	0,25	0,25	0,25
5	k-NN	0,25	0,25	0,25	0,25
10	k-NN	0,25	0,26	0,25	0,14
All	k-NN	0,25	0,25	0,25	0,25
5	Bayes	0,26	0,27	0,26	0,15
10	Bayes	0,26	0,29	0,26	0,14
All	Bayes	0,25	0,26	0,25	0,15
5	ANN	0,25	0.16	0,25	0,1
10	ANN	0,25	0,25	0,19	0,18
All	ANN	0,25	0,25	0,25	0,23

TABLE 3. Experiments for Slowhttptest.

Features	**Classifier**	**Accuracy rate**	**Recall**	**Precision**	**F1-score**
5	Decision tree	0,39	0,39	0,43	0,4
10	Decision tree	**0,52**	**0,53**	**0,55**	**0,54**
All	Decision tree	**0,53**	**0,54**	**0,57**	**0,55**
5	Random forest	0,4	0,4	0,43	0,41
10	Random forest	0,5	0,5	0,53	0,51
All	Random forest	0,51	0,54	0,55	0,54
5	k-NN	0,4	0,51	0,43	0,45
10	k-NN	0,42	0,52	0,45	0,47
All	k-NN	0,38	0,49	0,41	0,44
5	Bayes	0,38	0,5	0,41	0,33
10	Bayes	0,37	0,41	0,39	0,32
All	Bayes	0,35	0,23	0,39	0,27
5	ANN	0,38	0,5	0,41	0,34
10	ANN	0,35	0,25	0,38	0,27
All	ANN	0,37	0,55	0,4	0,34

When Table 3 is interpreted, it is concluded that the best result is obtained from the decision tree in classification with all attributes. Approaches other than random forest was not very successful. When the remaining algorithms are ranked, k-NN, ANN and finally Bayes come.

When the number of features is reduced to 5, the results are again quite low. k-NN gave the best outcomes in 5 attributes. The random forest algorithm has more or less the same outputs. However, it is seen that other algorithms are ineffective with 40% and below results. Here again, Bayes (with a slight difference from ANN) has been the most abortive algorithm, as in all attributes.

The number of features was reduced to 10 and the experiments were repeated. In 10, outputs closer to the experiments with all features were obtained. Although the outcomes are not very good, there is not enough amount of data for Slowhttptest. Therefore, results of 50% and above are acceptable. The best approach was taken as the decision tree. When other approaches are analyzed, it can be concluded that they provide similar performance in random forest. Next comes the k-NN algorithm, Bayes and finally ANN. Contrary to all features and 5 features, Bayes was more effective here than ANN. 10 most effective features selected by GA in classifying Slowhttptest are:

- “Bwd IAT Mean”
- “Flow Bytes/s”
- “RST Flag Count”
- “Bwd IAT Max”
- “Fwd Packet Length Min”
- “Total Fwd Packet”
- “Flow Duration”
- “Bwd PSH Flag”
- “Active Min”
- “FWD Init Win Bytes”

When Table 4 is evaluated, it is deduced that the most effective algorithms are decision tree and random forest when all metrics are taken into account. Values of 96% and above are quite good results. When the outcomes of other approaches are examined, k-NN, Bayes and finally ANN come respectively.

When the experimental results for feature selection are examined, the random forest algorithm comes to the fore, similar to other experiments. Likewise, the success of the decision tree draws attention. Considering the 0.95 and above performance, it is deduced that the GA can predict the 5 most effective features in classification:

- “Bwd IAT Total”
- “Active Max”
- “Bwd IAT Max”
- “Fwd IAT Std”
- “Bwd Packet Length Std”

TABLE 4. Experiments for Slowloris.

Features	**Classifier**	**Accuracy rate**	**Recall**	**Precision**	**F1-score**
5	Decision tree	0,96	0,94	0,94	0,94
All	Decision tree	**0,97**	**0,96**	**0,96**	**0,96**
5	Random forest	**0,97**	**0,96**	**0,96**	**0,96**
All	Random forest	**0,97**	**0,96**	**0,95**	**0,96**
5	k-NN	0,7	0,5	0,61	0,54
All	k-NN	0,7	0,5	0,61	0,54
5	Bayes	0,62	0,41	0,5	0,4
All	Bayes	0,61	0,5	0,33	0,37
5	ANN	0,62	0,41	0,5	0,4
All	ANN	0,55	0,33	0,4	0,31

After the decision tree and random forest comes k-NN, then ANN and finally Bayes. In Figure 7, the accuracy rates before GA application (with all attributes) and accuracy rates after GA application (with feature selection) are shown according to the values in Tables 1, 2, 3 and 4. Here, considering all the outputs, it can be said that the GA was successful.

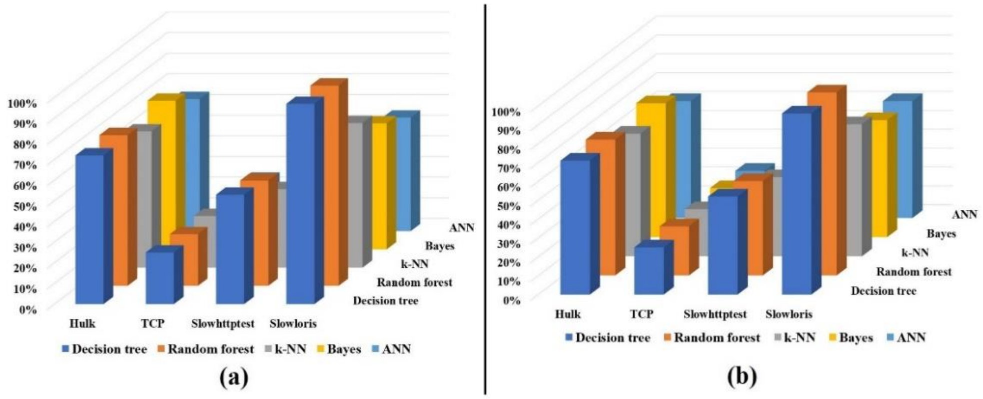


FIGURE 7. Accuracy rate (a) without GA (including all features) (b) with GA.

TABLE 5. Experimental results for all categories.

Features	**Classifier**	**Accuracy rate**	**Recall**	**Precision**	**F1-score**
10	Decision tree	0,54	0,58	0,56	0,55
All	Decision tree	0,58	0,61	0,6	0,59
10	Random forest	**0,55**	**0,59**	**0,57**	**0,56**
All	Random forest	**0,58**	**0,64**	**0,6**	**0,58**
10	k-NN	0,45	0,44	0,44	0,38
All	k-NN	0,48	0,42	0,46	0,41
10	Bayes	0,44	0,35	0,42	0,32
All	Bayes	0,41	0,34	0,39	0,28
10	ANN	0,28	0,23	0,25	0,21
All	ANN	0,43	0,41	0,4	0,33

When the experiments with all groups from Table 5 are examined, it is inferred that the random forest algorithm is the best algorithm in both all features and 10 features, as in the results separately. The decision tree also produced approximate results with the random forest. While there is k-NN, ANN and Bayes order in the remaining algorithms in terms of all attributes, the ranking in terms of 10 attributes is k-NN, Bayesian and ANN. The 10 most effective features estimated (for the random forest classifier) are as follows:

- “Bwd Header Length”
- “Fwd Segment Size Avg”
- “Fwd IAT Total”
- “URG Flag Count”
- “Bwd Packet Length Min”
- “Bwd Packet Length Mean”
- “Bwd IAT Std”
- “Fwd IAT Mean”
- “Bwd PSH Flags”
- “Bwd IAT Min”

It is expected that the results are in average values when the experiments are considered separately. The accuracy rate is lowered due to Slowhttptest and especially TCP. In separate tests for TCP, outputs are around 25%. The poor results in separate experiments (although the test was performed with more data) will cause much lower values when the number of classes increases and the number of data decreases. In the experiments for Slowloris, it was observed that the results were quite good, although there were few samples. Unlike TCP, this category will increase the overall accuracy rate.

4. Conclusion

Technological developments have brought many innovations that make life easier, as well as operations that are used for malicious purposes. It is very important to keep data safe against the online fraud. Therefore, strict precautions are taken and systems are developed to prevent attacks. The effective performance of artificial intelligence-based approaches in making inferences has increased their applicability to problems in real life. Machine learning is a subfield of artificial intelligence and its algorithms are widely utilized.

In this study, classification was made by applying decision tree, random forest, k-NN, Bayesian and ANN approaches on the USB-IDS-1 dataset, which was prepared for attack detection. Then, feature reduction was done with GA and its performance was evaluated. The dataset contains rows of 83 attributes belonging to 4 different attack types (Hulk, TCP, Slowhttptest, Slowloris). However, the data ratio between groups is highly uneven (Figure 1). If experiments are performed in this way, the algorithms will classify according to the dominant group. In this case, it causes misinterpretation of performances. Therefore, separate experiments were conducted for each group. An equal number of lines were taken from each group, taking into account the distribution within itself. Nevertheless, the performances of all groups and algorithms were examined. For this, an equal number of samples (according to the lowest amount of data.) were taken from each class and experiments were carried out.

In the light of experimental results, the decision tree and random forest algorithm were prosperous in the Hulk and Slowloris dataset. Especially in Slowloris, they performed quite well with 95% and above outputs. In general, it is concluded that these two algorithms are more effective than the others in all groups. In the Slowhttptest data, these two algorithms showed average performance, however the other methods again yielded worse results. All algorithms have failed to classify TCP data in terms of both full features and reduced features. When the results including all groups were examined, it was seen that the best values were between 55% and 64%. This is an expected outputs considering the separate classifications.

In future studies, experiments can be performed with different optimization algorithms (Particle Swarm Optimization (PSO) [24], Artificial Bee Colony (ABC) [25], etc.) and different data sets. In addition, the performance of different classification methods and deep learning approaches (Convolutional Neural Networks (CNN)), which are advanced versions of ANN, can be evaluated.

Author Contribution Statements The authors contributed equally to this study.

Declaration of Competing Interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kaplan, A., Haenlein, M., Siri, Siri, in my hand: Who"s the fairest in the land? on the interpretations, illustrations, and implications of Artificial Intelligence, *Bus. Horiz.*, 62 (1) (2019), 15-25, https://doi.org/10.1016/j.bushor.2018.08.004.

[2] Samuel, A. L., Some studies in machine learning using the game of checkers, *IBM J. Res. Dev.*, 3 (3) (1959), 210-229, https://doi.org/10.1147/rd.33.0210.

[3] Aburomman, A. A., Reaz, M. B. I., Ensemble of binary SVM classifiers based on PCAand LDA feature extraction for intrusion detection, *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, (2016), 636-640.

[4] Al-Jarrah, O. Y., Al-Hammdi, Y., Yoo, P. D., Muhaidat, S., Al-Qutayri, M. Semi-supervised multi-layered clustering model for intrusion detection, *Digit. Commun. Netw.*, 4 (4) (2018), 277-286.

[5] Al-Yaseen, W. L., Othman, Z. A., Nazri, M. Z. A. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system, *Expert Syst. Appl.*, 67 (1) (2017), 296-303.

[6] An, X., Su, J., Lü, X., Lin, F., Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system, *EURASIP JWCN*, 249 (1) (2018), 1-9.

[7] Belavagi, M. C., Muniyal, B., Performance evaluation of supervised machine learning algorithms for intrusion detection, *Procedia Comput. Sci.*, 89 (1) (2016), 117-123.

[8] KDD, The 1999 KDD intrusion detection, 1999, http://kdd.ics.uci.edu/databases/kddcup99/task.html.

[9] Hick, P., Aben, E., Claffy, K., Polterock, J., The CAIDA DDoS attack 2007 dataset, 2007.

[10] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A., A detailed analysis of the KDD CUP 99 data set, in 2009 *CISDA*, (2009), 1-6.

[11] Catillo, M., Del Vecchio, A., Ocone, L., Pecchia, A., Villano, U., USB-IDS-1: a public multilayer dataset of labeled network flows for IDS evaluation, *51st Annual IEEE/IFIP DSN-W*, (2021), 1-6, https://doi.org/10.1109/DSN-W52860.2021.00012.

[12] Holland, J. H., Genetic algorithms, *Sci. Am.*, 267 (1) (1992), 66-73.

[13] Catillo, M., Pecchia, A., Villano, U., Measurement-based analysis of a DoS defense module for an open source web server, *Testing Software and Systems: 32nd IFIP WG 6.1 International Conference, ICTSS,* (2020), 121-134.

[14] Catillo, M., Del Vecchio, A., Pecchia, A., Villano, U., Transferability of machine learning models learned from public intrusion detection datasets: the CICIDS2017 case study, *Softw. Qual. J.*, (2022), 1-27.

[15] Kalutharage, C. S., Liu, X., Chrysoulas, C., Explainable AI and deep autoencoders based security framework for IoT network attack certainty, *Lect. Notes Comput. Sci.*, (2022), 13745, https://doi.org/10.1007/978-3-031-21311-3_8.

[16] Russell, S. J., Norvig, P., Artificial Intelligence a Modern Approach, Pearson Education, Inc., New York, 2010.

[17] Breiman, L., Random forests, *Mach. Learn.*, 45 (2001), 5-32.

[18] Cover, T., Hart, P., Nearest neighbor pattern classification, *IEEE Trans. Inf. Theory*, 13 (1) (1967), 21-27.

[19] Li, X., Ye, N., Decision tree classifiers for computer intrusion detection, *In Real-Time System Security*, (2003), 77-93.

[20] Ozsari, S., Uguz, H., Hakli, H., Implementation of meta-heuristic optimization algorithms for interview problem in land consolidation: A case study in Konya/Turkey, *Land Use Policy*, 108 (2021), 105511.

[21] Google colab., (2023). Available: https://research.google.com/colaboratory/faq.html. [Accessed: May 2023].

[22] Ozsari, S., Yapicioglu, F. R., Yilmaz, D., Kamburoglu, K., Guzel, M. S., Bostanci, G. E., Acici, K., Asuroglu, T., Interpretation of magnetic resonance images of temporomandibular joint disorders by using deep learning, *IEEE Access*, 11 (2023), 49102-49113, https://doi.org/10.1109/ACCESS.2023.3277756.

[23] Kingma, D. P., Jimmy, Ba., Adam: a method for stochastic optimization, arXiv:1412.6980, 2014.

[24] Kennedy, J., Eberhart, R., Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, 4 (1995), 1942-1948, https://doi.org/10.1109/ICNN.1995.488968.

[25] Karaboga, D., An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.