Anadolu Üniversitesi Bilim ve Teknoloji Dergisi A- Uygulamalı Bilimler ve Mühendislik
Anadolu University Journal of Science and Technology A- Applied Sciences and Engineering

ANADOLU UNIVERSITY

2018 - Volume: 19 Number: 1
Page: 206 - 218
DOI: 10.18038/aubtda.356705
Received: 21 November 2017 Revised: 02 February 2018 Accepted: 06 February 2018

FEATURE SELECTION AND COMPARISON OF CLASSIFICATION ALGORITHMS FOR INTRUSION DETECTION

Sevcan YILMAZ GÜNDÜZ [1,*], Muhammet Nurullah ÇETER [1]

[1] Computer Engineering, Faculty of Engineering, Anadolu University, Eskişehir, Turkey

ABSTRACT

The increase in the frequency of use of the Internet causes the attacks on computer networks to increase. Such phenomena also increase the importance of intrusion detection systems. In this paper, KDD Cup 99 dataset is used for the classification of the network attacks. Four different classification algorithms were used, and the results were compared. These algorithms were multilayer perceptron network, decision trees, fuzzy unordered rule induction algorithm (FURIA) and support vector machines. The most successful algorithm in this dataset found as FURIA. As the second part of this study, the most important feature sets were found by correlation-based feature selection and best first search algorithm. Then, the results of classification algorithms were compared with these new feature sets according to the performance of the algorithms.

Keywords: KDD Cup 99 Dataset, Support vector machines, FURIA, Intrusion detection system

1. INTRODUCTION

Recently, attacks on the computer networks are increasing with the spreading use of the internet. These attacks are carried out in different forms. Attackers surfing the internet find different exploits of systems and attack the systems in a variety of ways. An attacker can steal the information found on the institution or personal computers. Attacks can cause significant problems in Internet-based services. These cyber-attacks on institutions and people are negatively affecting the image of institutions and people. Intrusion detection systems (IDS) have been developed to prevent these attacks on computer networks. Intrusion detection systems monitor all network traffic and identify suspicious situations in incoming and outgoing connections. Different methods such as statistics, artificial intelligence, and data mining have been used in intrusion detection systems. Intrusion detection systems are divided into two primary groups as signature-based intrusion detection and anomaly-based intrusion detection. Signature-based intrusion detection systems detect known attacks, while anomaly-based intrusion detection systems help detect unknown attacks. Intrusion detection systems also allow classification of attacks.

Several articles have been published in the literature on network intrusion detection system [1-7]. In [1], an SVM based network intrusion detection system is proposed. This method combines hierarchical clustering, feature selection and Support Vector Machines (SVM). In [2] a genetic fuzzy system is designed to deal with intrusion detection problem. Experiments were performed in DARPA dataset. In [3], the results show that Hidden Naïve Bayes model gives a better performance than other state-of-the-art models in the classification of network attacks. In [4] correlation-based feature selection, information gain and gain ratio are used to reduce the features in IDS, and Naïve Bayes algorithm is used for classification. In [5], a combination of filters, discretizers are used to reduce the features in KDD Cup 99 dataset in order to classify the network attacks. In [6], a hybrid algorithm which combines k-means clustering with radial basis kernel function of SVMs to reduce features and classification of KDD Cup 99 dataset. In [7], logistic regression, Gaussian Naïve Bayes, SVMs and Random Forest algorithms are used for classification in NSL-KDD dataset.

*Corresponding Author: sevcany@anadolu.edu.tr

In this study, some classification algorithms were used to determine which attacks were better classified. These are multilayer perceptron, C4.5 decision tree algorithm, fuzzy unordered rule induction algorithm (FURIA) and SVMs. KDD Cup 99 (Knowledge Discovery and Data Mining Tools Competition) dataset were used for intrusion detection, and the performance of given algorithms are compared. As a second part of the study, feature selection operation was performed on that dataset and results were compared. The most successful feature sets are found for each algorithm.

2. INTRUSION DETECTION SYSTEMS

Any attempt to intimidate and corrupt the confidentiality, integrity, and accessibility of information is called as network attack. Today, there are many different attacks on information systems. Intrusion detection systems have been developed to prevent these attacks. Intrusion detection systems are designed to take precautions against the risk of attack on the network [8]. The intrusion detection system is divided into host based and network based on the environment. A host-based intrusion detection system only detects attacks on that host. Network-based intrusion detection system plays a role in detecting all attacks in that network. Intrusion detection systems can be divided into two groups as signature-based intrusion detection system and anomaly-based intrusion detection system according to the attack detection method. The structure of the anomaly based IDS and signature-based IDS are shown in Figure 1. Anomaly-based IDS detects unusual situations in the network traffic as attacks, and it can detect new attacks in the network. The anomaly-based IDS is more likely to give false warnings because it does not use any attack signatures. In anomaly-based IDS (Figure 1.a), servers and users go out on the internet, and their network statistics are stored in the database. In general, these network statistics are close to each other. However, sudden increases in network traffic according to network statistics database indicate that the intrusion detection system is an attack. The signature-based IDS compares it with the signatures in the database and decides whether or not it is an attack as seen in Figure 2.1.b. Signature-based IDSs detect only known attacks, and they cannot detect new attacks that do not exist in the database [8].

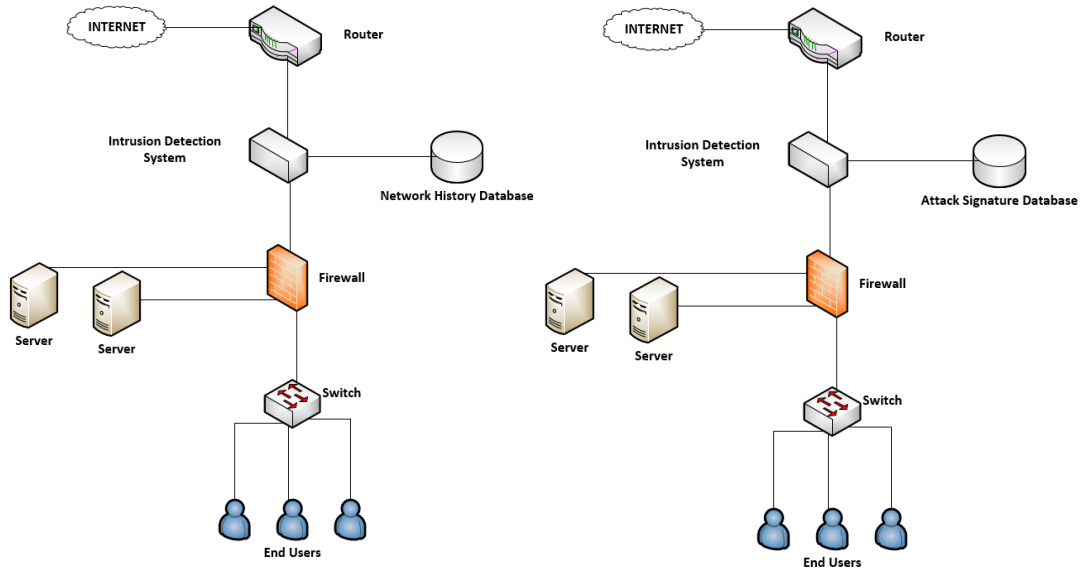

Figure 1. a) Anomaly-Based Intrusion Detection System **b)** Signature-Based Intrusion Detection System [8]

3. THE ALGORITHMS USED IN CLASSIFICATION OF NETWORK ATTACKS

3.1. Multilayer Perceptron

Neural networks are designed to model the human brain. Neural networks have nonlinearity, input-output mapping, adaptively, evidential response, contextual information, fault tolerance properties [9].

Neural networks are used in different application areas such as pattern recognition, time series prediction, signal processing, and control. Various types of neural network models are designed in the literature such as multilayer perceptron, radial basis functions, self-organizing maps. In this paper, we used multilayer perceptron (MLP) to find the type of the attack in intrusion detection system. The structure of the multilayer perceptron shown in Figure 2. The first layer is input layer that transmits the input signals to the hidden layer. The last layer is output layer that calculates the overall output of the system. The layers between the input layer and the output layer are called as a hidden layer. There can be more than one hidden layers.

Different activation functions can be used in MLP networks. These may be sigmoid and tangent hyperbolic functions. While learning unknown parameters in MLP, feedforward, and backpropagation algorithms are used. The output of the multilayer perceptron can be calculated in feed forward phase as follows:

$$y_k = F_k[\sum_{m=1}^{n_h} b_{k,m} f_m(\sum_{l=1}^{n_x} w_{m,l} x_l + w_{m,0}) + b_{k,0}] \quad (1)$$

where x, y, w, and b are input, output, weight and bias respectively.

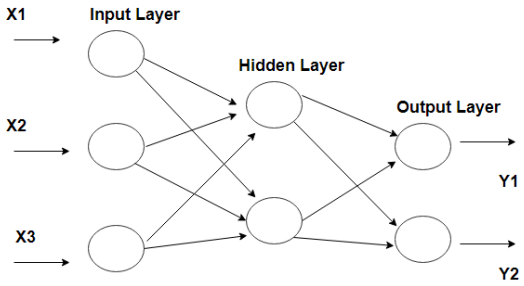

Figure 2. Multilayer Perceptron Neural Network

3.2. C4.5 Decision Tree Algorithm

Decision tree learning is one of the essential classification algorithms, and they are commonly used in data mining applications [10]. The reasons why decision trees are used very often are the speed of training and testing, the ease of interpretation, and the goodness of the visualization.

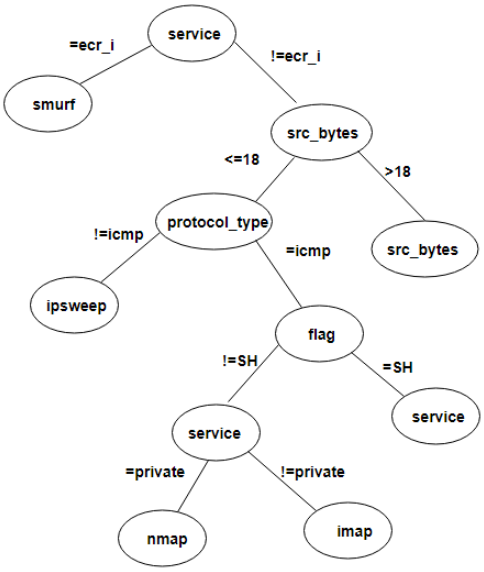

Figure 3. Decision tree example

In Figure 3, the tree structure used in the C4.5 algorithm is expressed on a sample. In decision tree algorithm, the highest value of entropy value is written at the top of the decision tree, and so the decision tree continues to be drawn. The entropy equation is given in Eq. 2.

$$H(x) = E(I(X)) = \sum_{i=1}^{n} p(x_i) \log_2 (1/\, p(x_i)) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad (2)$$

Here, $p(x_i)$ is the value that indicates the frequency of any feature.

3.3. Fuzzy Unordered Rule Induction Algorithm

FURIA is a fuzzy rule-based classification algorithm that extends the RIPPER algorithm [11]. Comprehensive and straightforward rule sets can be learned with this algorithm. In FURIA algorithm, unordered rule sets are learned instead of traditional rule lists. In this algorithm, rules have soft boundaries have again to be turned into crisp boundaries. Thus, this gives an opportunity to have a more flexible fuzzy design. FURIA learns namely a set of rules for each class in a one-vs-rest scheme and learned model might not be complete [11].

3.4. Support Vector Machines

Support vector machines are one of the most used machine learning algorithms in classification. It is possible to distinguish two groups by drawing a decision boundary between two groups, which are different from each other in classification [12, 13]. The decision boundary is shown in Figure 4. This boundary should be the farthest away from the members of the two groups. The support vector machines algorithm helps in choosing this decision boundary. Different kernel functions can be used in support vector machines. They can be linear, polynomial, Gaussian, sigmoid. In this paper, the Gaussian function was used for core functions in support vector machines.

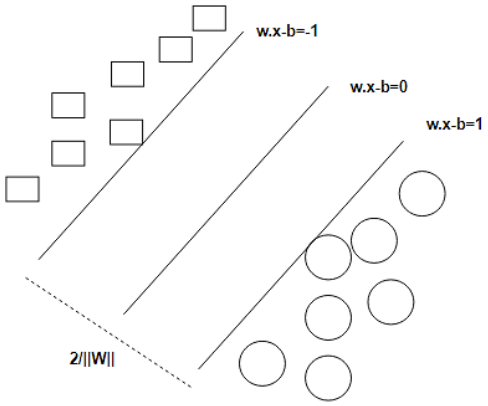

Figure 4. Linear support vector machines

The w * x - b is separator plane shown in Figure 4. The term $b / \|w\|$ gives the distance difference between two groups. This difference in distance is also called margin. If the margin is larger than classification performance is better than smaller margin.

4. KDD Cup 99 DATASET

KDD Cup 99 dataset [14] is one of the most used datasets in intrusion detection systems. At the same time, this data set is also known as the largest dataset used in intrusion detection systems. There are

close to 5 million records in this data set, and each record has 41 properties [15]. Some of these properties are numeric, and some are nominal. These features can be divided into 3 main classes: content features, server-based traffic features, and time-dependent traffic features. The 24 types of attack types in this dataset are divided into four main classes. Experiments were performed on 5000 records selected from the dataset. The attack types can be divided into four groups [5-16]:

Probe: This attack is to find open ports by scanning ports of a server or any computer. Thus, with these open ports, an attacker can easily attack these devices. Examples of this attack include ipSweep and portsweep. "ipSweep" is an attack that scans a certain port continuously. "portSweep" is an attack that scans all ports to find services on a server.

Denial of Service-DoS: These attacks are usually made to make the server out of service by sending a large number of requests to the server. Such attacks can be done by a single machine or by a large number of computers that are under control. Such machines that are under control are called zombie computers."smurf" is an example of DoS. "Smurf" is that ICMP packets are broadcast over the entire network.

Remote to Local-R2L: Remote to Local-R2Lis unauthorized access as a guest or as a user in the absence of user rights. "guess_passwd" attack is to enter the system by finding the insecure passwords.

User to root-U2R: An average user who does not have administrator rights take administrative exploits using some explanations in this attack types. "rootkit" is an example of U2R attack type. "rootkit" is a collection of programs that enable administrator-level access to a computer.

5. EXPERIMENTAL STUDY AND RESULTS

In this paper, 5000 entries from KDD Cup 99 dataset were received, and the experiments were done on this dataset. The types, categories and number of attacks used in the experiments are shown in Table 1.

Table 1. Attack types, categories and the number of attacks used in experiments

Attack Type	Category of the Attack	Number of the Attacks
normal	-	788
back	Dos	364
guess_passwd	R2L	50
imap	R2L	16
ipsweep	Probe	462
neptune	Dos	840
nmap	Probe	16
portsweep	Probe	5
rootkit	U2R	10
satan	Probe	462
smurf	Dos	1976
teardrop	Dos	4
warezclient	R2L	7
Total		**5000**

First of all, the experiments are done by using all 41 features. 10-fold cross validation applied to all algorithms. The results of the four algorithms, FURIA, decision tree, SVM and MLP are compared according to percentage of correctly classified samples. The results are shown in Table 2 according to all features used. The results are shown that FURIA gives better results than other three algorithms in intrusion detection. The number of correctly classified samples and the actual number of class members according to classes are shown in Figure 5. In addition, the confusion matrices for FURIA, C4.5 decision

tree, SVM and MLP algorithms is shown in Table 3, Table 4, Table 5 and Table 6 respectively. When the confusion matrices was examined, it was seen that the success rate was lower in the class with fewer samples such as 'warezclient' and 'rootkit'. The success rate in other classes is good according to confusion matrix. If we increase the samples in warezclient' and 'rootkit' classes, we can also take better results in that classes. The elapsed time of running the algorithms is given in Table 7. All the experiments are performed on a standard PC machine running 64-bit Windows 10 Enterprise operating system that has a 64 bit Intel(R) Core(TM) i5-3210M CPU 2.50 GHz processor and 4 GB of RAM. According to Table 7, the fastest algorithm is the decision tree algorithm, while the slowest algorithm is MLP.

Table 2. Comparison of algorithms according to all features used

Algorithm	**Correctly Classified Samples - Percentage**
FURIA	4965 - 99.3 %
C4.5	4950 - 99 %
SVM	4856 - 97.12%
MLP	3604 - 72.08%

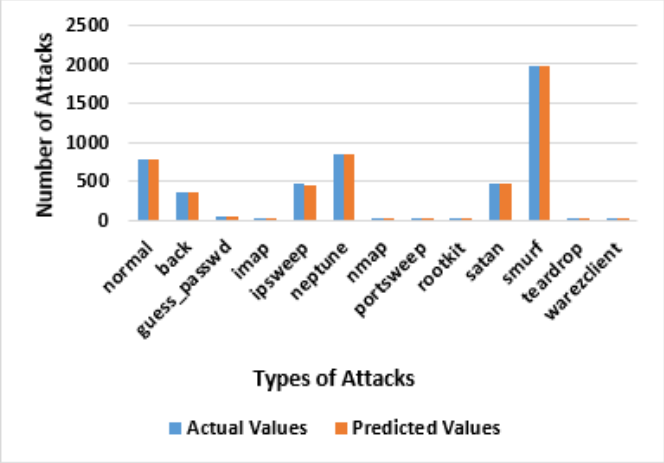

Figure 5. Distribution graph of actual and predicted values of classes according to FURIA algorithm for all properties

Table 3. The confusion matrix for the FURIA algorithm according to all the features used

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	778	1	0	0	3	0	0	0	1	3	1	0	1
b=back	1	362	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	1	0	48	0	1	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	3	0	0	0	458	0	0	0	0	0	1	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	1	0	0	0	0	0	0	4	0	0	0	0	0
i=rootkit	3	0	1	0	2	0	0	0	3	0	0	0	1
j=satan	2	0	0	0	1	0	0	0	0	459	0	0	0
k=smurf	0	1	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	1	0	0	0	0	0	0	0	0	0	0	3	0
m=warezclient	4	0	0	0	0	0	0	0	0	0	0	0	3

Table 4. The confusion matrix for the C4.5 algorithm according to all the features used

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	772	0	1	0	3	0	0	1	2	4	1	1	3
b=back	3	360	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	2	0	48	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	4	0	0	0	457	0	0	1	0	0	0	0	0
f=neptune	0	0	0	0	0	839	1	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	1	0	0	0	1	0	0	3	0	0	0	0	0
i=rootkit	6	0	0	0	3	0	0	0	1	0	0	0	0
j=satan	4	0	0	0	0	0	0	1	0	457	0	0	0
k=smurf	1	0	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	2	1	0	0	0	0	0	0	2	0	0	0	2

Table 5. The confusion matrix for the SVM algorithm according to all the features used

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	15	349	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	4	0	46	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	14	0	0	0	0	0	0	0	0	0
e=ipsweep	14	0	0	0	448	0	0	0	0	0	0	0	0
f=neptune	5	0	0	0	0	826	0	0	0	9	0	0	0
g=nmap	3	0	0	0	0	0	13	0	0	0	0	0	0
h=portsweep	1	0	0	0	0	0	0	1	0	3	0	0	0
i=rootkit	8	0	0	0	2	0	0	0	0	0	0	0	0
j=satan	37	0	0	0	0	21	1	0	0	403	0	0	0
k=smurf	8	0	0	0	0	0	0	0	0	0	1968	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	7	0	0	0	0	0	0	0	0	0	0	0	0

Table 6. The confusion matrix for the MLP algorithm according to all the features used

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	0	364	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	49	0	0	0	0	1	0	0	0	0	0	0	0
d=imap	16	0	0	0	0	0	0	0	0	0	0	0	0
e=ipsweep	462	0	0	0	0	0	0	0	0	0	0	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	16	0	0	0	0	0	0	0	0	0	0	0	0
h=portsweep	5	0	0	0	0	0	0	0	0	0	0	0	0
i=rootkit	10	0	0	0	0	0	0	0	0	0	0	0	0
j=satan	462	0	0	0	0	0	0	0	0	0	0	0	0
k=smurf	0	0	0	0	0	0	0	0	0	0	1976	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	6	0	0	0	0	0	0	0	0	0	1	0	0

Table 7. The performance of algorithms over time

Algorithm	Time (sn)
C4.5	0.8
FURIA	16.7
SVM	56.6
MLP	244.9

5.1. Feature Selection in KDD Cup 99 Dataset

In this part, feature selection operation is done on KDD Cup 99 dataset. Correlation-based feature selection (CFS) [4, 17, 18]was used as the property evaluator, and best first search (BFS) is used as search method. In CFS, the algorithm selects highly correlated feature sets. This algorithm is based on the hypothesis: "Good feature subsets contains features highly correlated with the class, yet uncorrelated with each other" [18]. The feature selection is good in terms of extracting unnecessary information from data sets and speeding up the model. Correlation between feature subset M_s can be calculated as follows [4, 18] :

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3}$$

where k is the number of feature subset, $\overline{r_{cf}}$ is mean of the correlation between feature subset and class features and $\overline{r_{ff}}$ is the mean correlation between features.

The KDD Cup 99 dataset contains 41 features. Therefore, we can have $2^{41} - 1$ (2,199,023,255,551) different feature sets. By using correlation-based selection and BFS algorithm, 11 features are selected, and they are ranked in Table 8 according to importance.

Table 8. Selected features using BFS Algorithm

Rank	Feature Name
1	service
2	src_bytes
3	dst_bytes
4	wrong_fragment
5	count
6	srv_count
7	diff_srv_rate
8	dst_host_diff_srv_rate
9	dst_host_same_src_port_rate
10	dst_host_srv_diff_host_rate
11	dst_host_rerror_rate

10-fold cross validation is applied in all algorithms in the feature selection part. According to Table 8, the most important feature is "service." When we use only "service" feature, the classification results are shown in Table 9 and the most successful algorithm is decision tree algorithm at this time. In addition, different tests are done with subsets of the feature set. The percentages of the correctly classified sample are shown in Table 9 for different feature sets. According to these test results, the most successful feature sets and success rate for FURIA, SVM, decision tree and MLP are shown in Table 10. When we compared the best results (Table 10) with the results using all the features (Table 2), MLP results were found to be quite different from the other algorithms. MLP gives its best results with 4 features. When the number of features increases, MLP algorithm becomes more difficult to learn. Therefore, MLP gives worse results with all 41 features used. In addition, the other algorithms, FURIA, decision tree, and SVM gives their best results with different feature sets. After reviewing the results of the feature selection process, it is not necessary to use all 41 features to find the best results. The

distribution graph for the C4.5 algorithm according to "service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate" count features used is shown in Figure 6. The confusion matrices of the best results according to best feature sets for FURIA, C4.5 decision tree, SVM and MLP are shown in Table 11, Table 12, Table 13 and Table 14 respectively.

Table 9. Results for different feature sets according to algorithms

Features	FURIA	C4.5	SVM	MLP
service	84.98%	85.68%	85.32%	82.08%
service, src_bytes	96.52%	96.36%	96.34%	81.16%
service, src_bytes, dst_bytes	96.68%	96.36%	96.5%	83%
service, src_bytes, dst_bytes, wrong_fragment	96.64%	96.36%	96.5%	83.2%
service, src_bytes, dst_bytes, wrong_fragment, count	98.22%	97.8%	97.2%	87.42%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count	98.34%	97.76%	97.28%	85.58%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate	99.02%	98.2%	97.34%	74.36%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate	99.48%	98.62%	97.44%	69.98%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate	99.58%	98.76%	97.44%	79.42%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate	99.46%	99.24%	97.44%	82.6%
service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_rerror_rate	99.5%	98.9%	97.44%	85.24%

Table 10. Best results and best feature sets for each algorithm

Algorithm	Correctly Classified Samples - Percentage	RMSE	Used Features
FURIA	4979-99.58%	0.0178	service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate
C4.5	4962 - 99.24%	0.0239	service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate
SVM	4872 -97.44 %	0.0472	service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate
MLP	4371- 87.42%	0.0903	service, src_bytes, dst_bytes, wrong_fragment, count

Table 11. The confusion matrix for the FURIA algorithm according to service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	786	0	0	0	0	0	0	0	0	2	0	0	0
b=back	1	363	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	0	0	50	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	1	0	0	0	459	0	0	0	0	0	1	0	1
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	2	0	0	0	0	0	0	0	0	3	0	0	0
i=rootkit	2	0	1	0	1	0	0	0	6	0	0	0	1
j=satan	2	0	0	0	0	1	0	0	0	459	0	0	0
k=smurf	1	0	0	0	0	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	2	0	0	0	0	0	0	0	0	0	0	0	5

Table 12. The confusion matrix for the C4.5 algorithm according to service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	774	2	0	0	2	0	0	0	2	5	1	0	2
b=back	3	361	0	0	0	0	0	0	0	1	0	0	0
c=guess_passwd	0	0	50	0	0	0	0	0	0	0	0	0	0
d=imap	0	0	0	16	0	0	0	0	0	0	0	0	0
e=ipsweep	2	0	0	0	455	0	0	0	0	0	5	0	0
f=neptune	0	0	0	0	0	840	0	0	0	0	0	0	0
g=nmap	0	0	0	0	0	0	16	0	0	0	0	0	0
h=portsweep	0	0	0	0	0	0	1	3	1	0	0	0	0
i=rootkit	1	0	0	0	1	0	0	0	7	0	0	0	1
j=satan	3	0	0	0	0	0	0	1	1	457	0	0	0
k=smurf	0	0	0	0	1	0	0	0	0	0	1975	0	0
l=teardrop	0	0	0	0	0	0	0	0	0	0	0	4	0
m=warezclient	1	0	0	0	0	0	0	0	2	0	0	0	4

Table 13. The confusion matrix for the SVM algorithm according to service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate

Predicted / **Real**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**
a=normal	788	0	0	0	0	0	0	0	0	0	0	0	0
b=back	13	351	0	0	0	0	0	0	0	0	0	0	0
c=guess_passwd	2	0	48	0	0	0	0	0	0	0	0	0	0
d=imap	2	0	0	14	0	0	0	0	0	0	0	0	0
e=ipsweep	5	0	0	0	457	0	0	0	0	0	0	0	0
f=neptune	3	0	0	0	0	828	0	0	0	9	0	0	0
g=nmap	0	0	0	0	16	0	0	0	0	0	0	0	0
h=portsweep	0	0	0	0	5	0	0	0	0	0	0	0	0
i=rootkit	8	0	0	0	2	0	0	0	0	0	0	0	0
j=satan	30	0	0	0	2	20	0	0	0	410	0	0	0
k=smurf	5	0	0	0	0	0	0	0	0	0	1971	0	0
l=teardrop	2	0	0	0	0	0	0	0	0	0	0	2	0
m=warezclient	4	0	0	0	0	0	0	0	0	0	0	0	3

Table 14. The confusion matrix for the MLP algorithm according to service, src_bytes, dst_bytes, wrong_fragment, count

Predicted / Real	a	b	c	d	e	f	g	h	i	j	k	l	m
a=normal	770	0	1	0	8	9	0	0	0	0	0	0	0
b=back	328	0	0	0	36	0	0	0	0	0	0	0	0
c=guess_passwd	28	0	5	0	12	5	0	0	0	0	0	0	0
d=imap	2	0	2	0	7	5	0	0	0	0	0	0	0
e=ipsweep	34	0	0	0	423	5	0	0	0	0	0	0	0
f=neptune	8	0	0	0	9	821	0	0	0	2	0	0	0
g=nmap	14	0	0	0	0	2	0	0	0	0	0	0	0
h=portsweep	4	0	0	0	0	1	0	0	0	0	0	0	0
i=rootkit	9	0	0	0	0	1	0	0	0	0	0	0	0
j=satan	33	0	1	0	16	35	0	0	0	377	0	0	0
k=smurf	0	0	0	0	1	0	0	0	0	0	1975	0	0
l=teardrop	4	0	0	0	0	0	0	0	0	0	0	0	0
m=warezclient	6	0	0	0	0	0	0	0	0	0	1	0	0

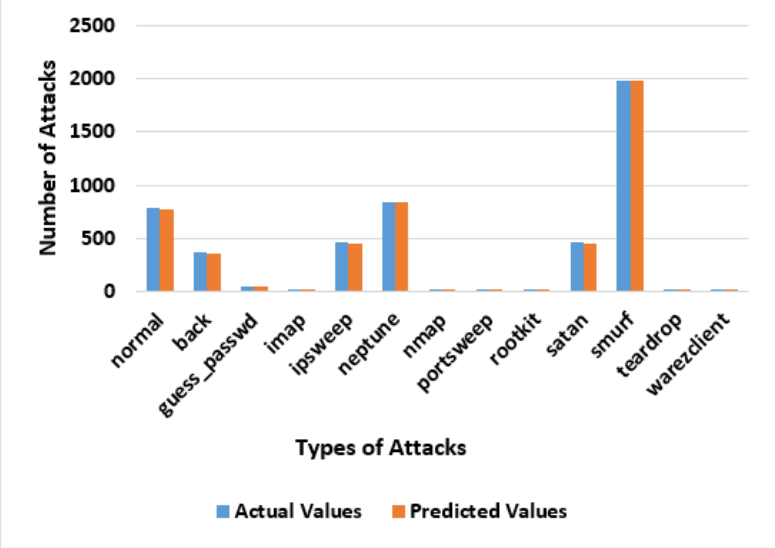

Figure 6. Distrubution graph of actual and predicted values of classes according to C4.5 algorithm for service, src_bytes, dst_bytes, wrong_fragment, count, srv_count, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate

6. CONCLUSION

In this study, the development of intrusion detection systems to protect cyber attacks against information systems is discussed. Until now, different algorithms are used in intrusion detection systems. C4.5, FURIA, MLP, SVM algorithms were used for intrusion detection systems for KDD Cup 99 dataset in this study. Before feature selection operation, the algorithms were compared and FURIA algorithm is found as the most successful algorithm. As the second part of the study, feature selection operation was done on KDD Cup 99 dataset. CFS was selected as a feature evaluator, and BFS algorithm is used as a search method. The most important 11 features were selected, and the subsets of this feature set were used for classification. So, the most successful feature set for each algorithm was found. Again, FURIA is the most successful algorithm, and MLP gives lower performance than other algorithms in this study. Finally, when the elapsed time of the algorithms is compared, the decision tree algorithm is given the quickest answer.

REFERENCES

[1] Horng SJ, Su MY, Chen YH, Kao TW, Chen RJ, Lai JL, et al. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 2011; 38: 306-313.

[2] Abadeh MS, Mohamadi H, Habibi J. Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. Expert Systems with Applications 2011; 38: 7067-7075.

[3] Koc L, Mazzuchi TA, Sarkani S. A network intrusion detection system based on a Hidden Naive Bayes multiclass classifier. Expert Systems with Applications 2012; 39 : 13492-13500.

[4] Mukherjee S, Sharma N. Intrusion Detection using Naive Bayes Classifier with Feature Reduction. In: 2nd International Conference on Computer, Communication, Control and Information Technology; 2012; vol. 4, pp. 119-128.

[5] Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A. Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. Expert Systems with Applications 2011; 38: 5947-5957.

[6] Ravale U, Marathe N, Padiya P. Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function. In: International Conference on Advanced Computing Technologies and Applications; 2015, pp. 428-435.

[7] Belavagi MC, Muniyal B. Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. In: Twelfth International Multi-Conference on Information Processing; 2016; Bangalore, India; pp. 117-123.

[8] Alamleh AH. Network Intrusion Classification Using Data Mining Techniques. Masters MSc, Zarqa University, Jordan, 2015.

[9] Haykin SS. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[10] Han J, Kamber M. Data mining: concepts and techniques. San Francisco: Morgan Kaufmann Publishers, 2001.

[11] Huhn J, Hullermeier E. FURIA: an algorithm for unordered fuzzy rule induction. Data Mining and Knowledge Discovery 2009; 19: 293-319.

[12] Cortes C, Vapnik V. Support-Vector Networks. Machine Learning 1995; 20: 273-297.

[13] Vapnik VN. Statistical learning theory. New York: Wiley, 1998.

[14] University of California. (1999, October 8). KDD Cup 1999 Data. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[15] Lin SW, Ying KC, Lee CY, Lee ZJ. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Applied Soft Computing 2012; 12: 3285-3290.

[16] Tavallaee M, Bagheri E, Lu W. A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications; 2009; Ottawa, ON, Canada.

[17] Witten IH, Frank E, Hall MA. Data Mining Practical Machine Leanrning Tools and Techniques. Morgan Kouffman.

[18] Hall MA. Correlation-based Feature Selection for Machine Learning. Department of Computer Science; The University of Waikato, 1999.