A PROPOSED MODEL FOR DIMENSIONALITY REDUCTION TO IMPROVE THE CLASSIFICATION CAPABILITY OF INTRUSION PROTECTION SYSTEMS

Hajar Elkassabi, Mohammed Ashour and Fayez Zaki

Department of Electronics & Communication Faculty of Engineering,
Mansoura University, Mansoura, Egypt

ABSTRACT

Over the past few years, intrusion protection systems have drawn a mature research area in the field of computer networks. The problem of excessive features has a significant impact on intrusion detection performance. The use of machine learning algorithms in many previous researches has been used to identify network traffic, harmful or normal. Therefore, to obtain the accuracy, we must reduce the dimensionality of the data used. A new model design based on a combination of feature selection and machine learning algorithms is proposed in this paper. This model depends on selected genes from every feature to increase the accuracy of intrusion detection systems. We selected from features content only ones which impact in attack detection. The performance has been evaluated based on a comparison of several known algorithms. The NSL-KDD dataset is used for examining classification. The proposed model outperformed the other learning approaches with accuracy 98.8 %.

KEYWORDS

NSL-KDD, Machine Learning, Intrusion Detection Systems, Classification, Feature Selection.

1. INTRODUCTION

Securing the network against all kinds of threats is an essential part of system security management. When the risks are increasingly increasing, safety systems need to be built to make them smarter than ever before. Regular security measures such as firewalls and antivirus cannot stop the growing number of complex attacks which take place over a network connection to the Internet. An additional safety layer was introduced as a solution to improve network security by the protection levels using intrusion protection systems (IPS). These can be viewed as additional protection measures focused on a framework of intrusion detection to avoid malicious attacks [1]. Through references, there were two main methods for detecting intrusions, one based on anomaly and the other based on signature [2]. In the first technique, the intrusion protection system searches for the data type outside the behaviour of the normal data type. When it finds this type of data, the attack protection system treats it as a potential attack. Anomalies in the data are detected by studying confirmed statistical behaviour. So, the difference from the natural flow is detected as an anomaly. Thus, it can express a possible intrusion within the network. One of the main advantages of this method is to contribute to identifying unknown attacks. This method can also detect data anomalies by detecting attack accurately through this mechanism with low false positives and negative warnings. One of the disadvantages of methods based on the detection of anomalies in the data is that its performance is affected negatively due to regularly changes that

occur in the network, so the normal traffic profile should be updated from time to time for avoiding this problem. On the other hand, signature-based detection, which can also be called abuse-based detection, is used to search between a list of signatures or interference patterns to detect malicious data. This type of detection works in addition to a regular update of its database. When an attack occurs, the signatures of these attacks are created. Signing known attacks helps detect future attacks. An advantage of these techniques is to analyse and detect known attacks in an accurate and effective manner that generate low false alarm. The problem with the existing signature-based methods is that zero-day attacks cannot be detected [3].

The method for detecting anomalies in the data set depends mainly on the appropriate choice of features or dimensionality. It is essential that appropriately chosen features or dimensions maintain accuracy of disclosure while performing calculations quickly. Dimensionality reduction is an effective method used to improve the overall performance of the intrusion prevention systems because this method reduces the number of features used to detect the intrusion to the lowest possible value. If the excluded features are ineffective, this will significantly improve the speed of implementation of the anomaly detection in the data set. It is essential that this increase in detection speed does not significantly affect detection accuracy for data anomalies. On the other hand, failure to specify the correct dimensionality for the data set means excluding important characteristics will reduce the operating speed and detection accuracy [4].

One of the suggested methods in research is machine learning to establish systems to detect infiltration into the computer network [5]. Many references have been pointed to the effectiveness of machine learning techniques for improving network classification. For intrusion detection-based machine learning techniques it is not advisable to use all the features in the data set. Because the application of all features adds a burden to the methods of calculations used. On the other hand, choosing the right features improves efficiency and reduces the time spent on learning. The relevant function is then used for further processing after this process [6]. The measurement of the performance of anomaly detection systems in the data must be based on use of the standard data set. The NDL-KDD Dataset was a popular data series on intrusion protection systems to test the validity of the methods proposed in this form of study. Many studies in this research area were conducted using the NSL-KDD data set [7].

In this paper a new feature selection technique for dimensionality reduction is proposed. The next section explains the related work. The problem statement is discussed in section 3. The proposed model is shown in Section 4. A research methodology is described in section 5. The visualization of the data set is described in Section 6. Configuration and setups are shown in section 7. Experiments and results are discussed in Section 8. Section 9 describes discussion of results and performance analysis. Section 10 explains the summary and future work.

2. RELATED WORKS

Over the last few decades, researchers carried out studies using the NSL-KDD dataset [8,9]. These studies concentrated on training and testing several machine learning algorithms as shown in Table (1).

Sabhani and Serpen [9] utilized decision trees (DT) algorithm and got high accuracy, however this technique did not do well with R2L and U2R attacks as they contain new attack types.
Dhanabal and Shantharajah [7] applied for classification of SVM, J48 and Naïve Bayes algorithms. Application of correlation feature selection increases the accuracy and reduces detection time.

Shrivastava, Sondhi and Ahirwar [10] presented the IDS framework which improves the classification performance based on machine learning algorithms.
Deshmukh, Ghorpade and Padiya [11] focused on increasing accuracy by using classifiers such as Naïve Bayes. Several pre-processing steps have been implemented on the NSL-KDD dataset as Discretization and Feature selection.

The performance of the NSL-KDD dataset was evaluated by Ingre and Yadav [12] using Artificial Neural Networks. Results applied based on several performance measures such as false positive rate, accuracy and detection rate and better accuracy was found. The proposed model achieved a higher detection rate compared with existing models.

Table 1. Overview of previous machine learning techniques for intrusion detection

Ref.	Algorithms	Dataset	Year
[6]	J48 PCA	NSL-KDD	2012
[5]	Random Forest J48 SVM CART Naïve Bayes	NSL-KDD	2013
[13]	J48	NSL-KDD	2014
[14]	LSSVM-IDS	KDDCUP99 NSL-KDD KYOTO2006+	2014
[7]	J48 SVM Naïve Bayes	NSL-KDD	2015
[15]	Naïve Bayes	NSL-KDD	2015
[16]	J48 Naïve Bayes	KDDCUP99 Kyoto2006+	2017
[17]	SVM-CART	KDDCUP99	2017
[18]	J48 Random Forest PART	NSL-KDD	2018
[19]	RIPPER PART C4.5	NSL-KDD	2018
[20]	SVM ANN	NSL-KDD	2019

3. PROBLEM STATEMENT

Access to the information via Internet, files are exchanged over a network, emails are sent and received with attachments and databases are now part of the daily routine of many people and businesses. Nearly all electronic communication is subject to the task of effectively managing the risks of today's cyber world to protect itself from malware attacks and hacking threats. The hackers use Security Vulnerabilities in computer networks for this mischievous assault and intrusion threats. A firewall may be used as a preventative measure. Yet only minimal security is

available from firewalls. Usually, a single firewall is mounted before a server to defend against external attacks. In the case of hackers who use fake packages that include a malicious program, the protection mechanism is compromised when the firewall is tricked by the mispackages. In addition, the firewall is useless if the hacking is performed inside the network by an insider. A main element of device security management is to protect the network against all sorts of attacks. Because the threats are growing exponentially, security systems must be designed to make them smarter than ever. The increasing number of complex attacks that take place over a network connection to the Internet cannot be stopped by regular security measures such as firewalls and antivirus. An additional layer of security has been proposed as a solution to enhance network security by increasing layers of protection using intrusion protection systems (IPS). They can be considered as additional safety measures that based on an intrusion detection system to prevent intentional attack [1].

4. PROPOSED MODEL

The proposed system is a combination of feature selection and machine learning algorithms. The process steps are shown in Figure 1.

In this paper we applied a new feature selection method that depends on dividing the contents in every feature to (Genes) using the NSL-KDD dataset. The results are compared before and after deleting unimportant genes in every feature. For the simulation we used python (3.7.3) and Weka tool. They have various machine learning algorithms and tools for data pre-processing, Clustering, Classification, Visualization and Data analysis. The experimental steps are

1. Import data set “train & test”
2. Pre-process step (Data conversion, Data correlation, Data scaling).
3. Run the classifier.
4. Evaluate results analysis & Compare the results

We will import NSL-KDD train data for pre-processing steps then feed it to the classifier to complete the learning process. The test data file will be pre-processed also with the same pre-processing steps. After that we will feed the system with these hidden data (test-data) for validating the learning rate of every classifier, therefore the classifier accuracy is calculated. From pre-processing step, we obtain a 16-feature subset based on a higher accuracy than other

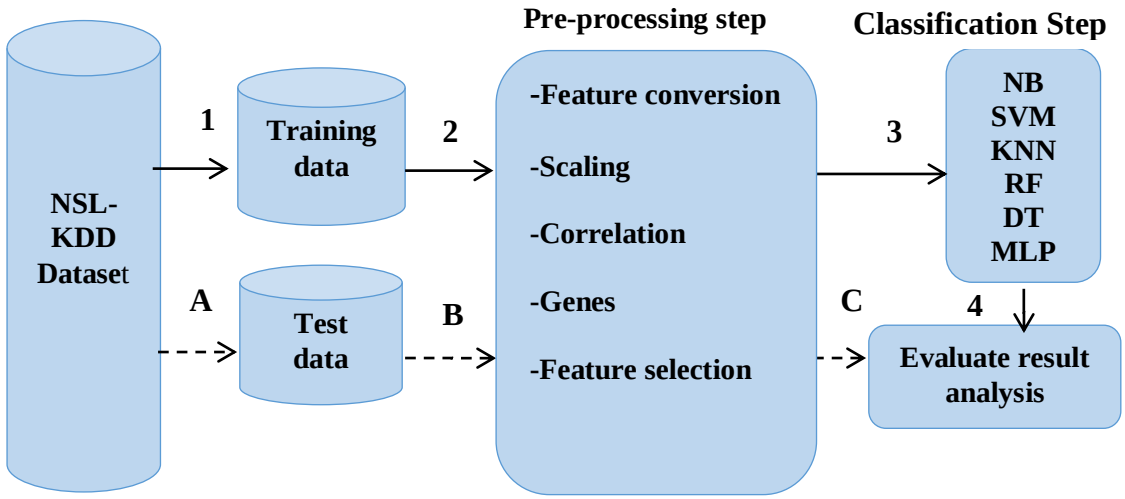

Figure 1. Our Proposed Model for Feature Selection & Classification

subsets. we discovered that not all features contents are important in attack detection. We named the feature content {Gene}. We will eliminate unimportant genes in every these 16-features. We choose only one effective gene in every feature depending on its frequency with attacks.

For example: Attribute 9 has values (Genes) {0,1,2,3}. The attack's symbols (A1 to A23) are represented in table (8). The new selection mechanism has been described in table (2) and figure (2), we notice that gene (0) has higher detection rate with 23 types of attacks which NSL-KDD contains, so we will choose it from this feature. We will do that for all feature in our subset. The experiments show that this model has higher accuracy compared with one which contains all feature's contents.

Table 2. Distribution of attacks with genes in feature (9)

Att	Gene	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
9	0	956	30	6	53	11	3599	18	9	7	41214	1493	67337	3	4	201	2931	9	3633	2646	2	892	890	20
9	1	0	0	1	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	0
9	2	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

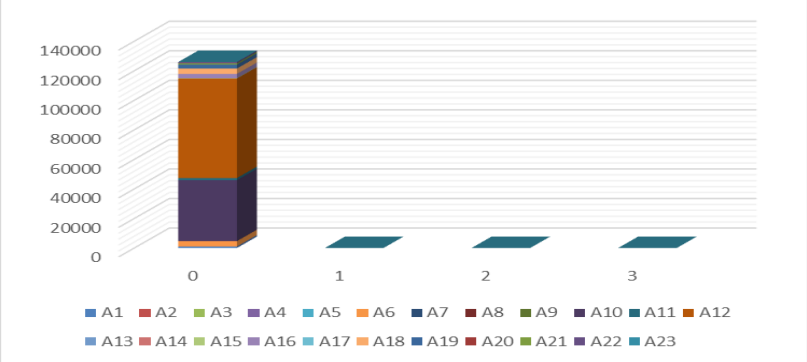

Figure.2 Genes Distribution in feature (9)

5. RESEARCH METHODOLOGY

A network is formed by packets that start and end at any time as data is transmitted from one source IP address to another target IP address under a certain protocol via transmission control protocol (TCP) systems. Every network is classified as regular or as an attack of exactly one specific type of attack. NSL-KDD Data Collection has been used in this paper, this dataset is a modified version of DARPA and KDD CUP99 managed by MIT Lincoln Labs. Nine weeks raw TCP dump data were obtained from Lincoln Labs for the local area network (LAN) pretending as a Typical US Air Force network [35]. The first seven weeks are data for the training set and the last two weeks is the test set. There are 42 variables in this dataset, one of which is the network condition, marked as an attack or normal. These research variables summarised into three categories as follows:

1) Essential features: all features collected from the TCP / IP are included in this group.
2) Traffic characteristics: this class describes the characteristics that are measured for a duration
3) Content features: we can evaluate functions like the number of failed logins attempts to recognize suspected behavior.

6. DATASET VISUALIZATION

(Network Security Laboratory Knowledge Discovery and Data Mining) NSL-KDD is extracted from the KDD dataset (the original version). The number of NSL-KDD features in each record is (42) while the last attribute explained the label or class. Each connection is labelled an attack type or normal [21]. The Total number of attacks presented in NSL-KDD are 39 attacks, each one of them is grouped in to four major classes:

1. DOS: denial-of-service, which means preventing legitimate users from accessing a service.
2. R2L: Remote-to-Local, which means accessing the victim machine by intruding into a remote machine.
3. U2R: User-to-Root, that means a normal account has been used to login in a victim network and attempt to get root privilege.
4. Probing: checking and scanning vulnerability on the victim machine for collecting data about it.
5.

As appeared in Table (3,4) the distributions of NSL-KDD dataset files, The NSL-KDD contains two files (training and testing). Test file includes different attacks which do not exist in the training file, it is significant to be noted.

Table 3. List of attacks presented in NSL-KDD

Attack Attribute	Attack Name
PROBE	Portsweep, Saint, Ipsweep, Satan, Nmap, Mscan. (6)
DOS	Back, Neptune, Processtable, Teardrop, Smurf, Apache2, Land, Mailbomb, Udpstorm, Pod. (10)
U2R	Rootkit, Buffer_overflow, Ps, Perl, Xterm, Loadmodule, Sqlattack, Httptunnel. (8)
R2L	Named, Warezmaster, Imap, Warezclient, Guess_Password, Snmpguess, Phf, Sendmail, Spy, Ftp_write, Xsnoop, Multihop, Snmpgetattack, Xlock, Worm. (15)

Table 4. NSL-KDD files Distributions

KDD dataset	Overall records	Dos	U2R	PROBE	R2L	Normal
KDD train	125973	45972	52	11656	995	67343
		36.46%	0.04%	9.25%	0.79%	53.46%
KDD test	22544	7458	200	2421	2754	9711
		33.08%	0.89%	10.74%	12.22%	43.07%

As it been clarified in figure (3&4), NSL-KDD dataset available in three versions:

a. KDDTrain+ with a total number of 125974 records.
b. KDDTrain+_20Percent which consists of 20% of the training data with 25192 records.
c. KDDTest+ with a total number of 22544 records.

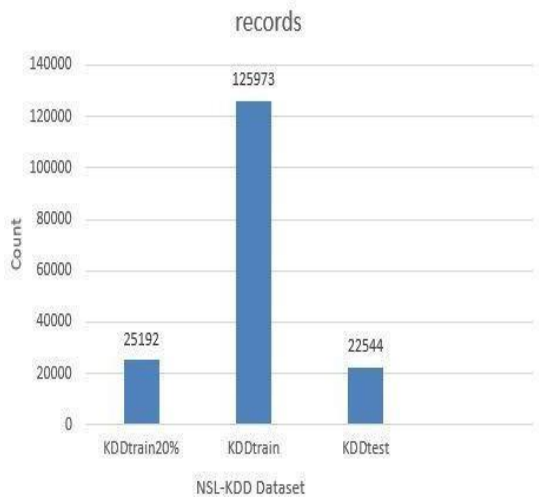

Figure 3. NSL-KDD dataset versions

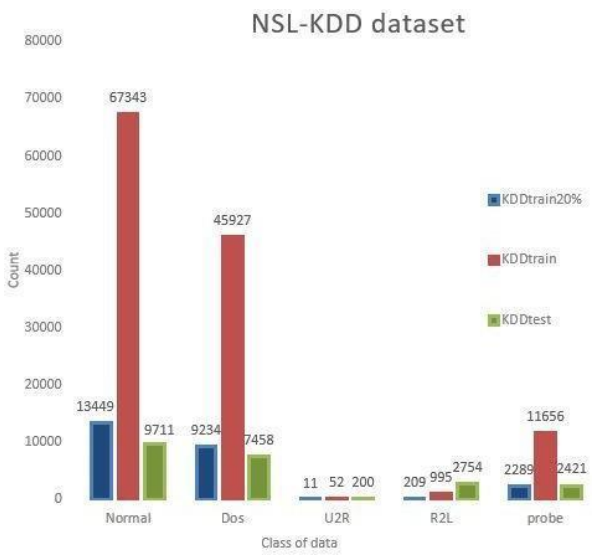

Figure 4. Statistics of NSL-KDD total records

Even though the NSL-KDD dataset had a few issues, it is an extremely successful dataset that can be utilized for research purposes [10], [22]. In addition, it is hard to acquire certifiable security datasets considering the idea of the security area and keeping in mind that there are other datasets.

7. SIMULATION TOOLS & SYSTEM CONFIGURATIONS

In feature selection step for obtaining (16-feature), we used Huffman coding by MATLAB. Huffman Coding is a lossless algorithm for data encryption [36]. The process underlying its system includes the sorting by frequency of numerical values. Using this code enable us to obtain frequencies between attacks and features for all instances in NSL-KDD.

The simulation tool used for the first and second experiments is python 3.7. Deep learning using multi-layer perceptron has been used in the third experiment using Waikato Environment For knowledge Analysis (Weka) version 3.8.3 by OS windows 10 enterprise Intel® Core™ i5-3230M CPU@ 2.60GH, (RAM) 6.00 GB.

8. EXPERIMENTS AND RESULTS

In this paper we divided our work into three experiments. In the first one we used many subsets (39-feature, 16-feature, and 4-features) for training & testing and five machine learning algorithms for classification (NB, KNN, SVM, DT, RF) then compared between them. A new feature selection model for enhancing classification accuracy has been discussed in the second experiment. The simulation tool used for the first and second experiments is python 3.7. Deep learning using multi-layer perceptron has been used in the third experiment using Weka as a simulation tool.

8.1. Performance Metrics

The following performance metrics have been used in our work

- True [Positive (TP): Record is exposed as an attack.
- True Negative (TN): Record Correctly identified as normal.

- False Positive (FP): When a classifier detected a normal record as an attack.
- False Negative (FN): a detector identifies an attack as a normal instance.
- F-measure. It is obtained from the following equations

$$precision = \frac{TP}{TP+FP} \quad (1)$$

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

$$F-measure = \frac{2\times precision\times recall}{recall+precision} \quad (3)$$

8.2. The First Experiment:

8.2.1. Pre-processing step

Pre-processing data is an important task for accuracy. This is because data is mostly noisy and sometimes has missing values, so feature selection or dimensionality reduction considered a major method of pre-processing which directly impact the accuracy of the model. Feature selection is the method of selecting some features out of the data and discarding the irrelevant ones [24]. NSL-KDD dataset contains training data which have 41 feature and class attribute that contain 23 type of attacks [7], after removing feature 20&21 because containing zeros we obtain 1st subset [39-feature]. We measure frequency between all features and 23 types of attacks, we found that features {9,11,13,15,21,22,23,24,27,28,29,30,31,37,40,41} have a highest effect in attack detection as shown in table (8), so these [16-feature] will be our 2nd subset. From [14],[13],[25],[26],[18],[27] we obtained a third subset which contains [4-features] [3,5,12,26] as common features in the previous researches.

Our pre-processing step has data visualization, data conversion & scaling and data correlation, we will do that for our three subsets.

A) Data visualization

As shown in Figure (9) Xattack column in train data contains different types of attacks, we will modify it so that it will have only Two unique values (attack & normal), as clarified for train data in the following figures (5,6,7,8)

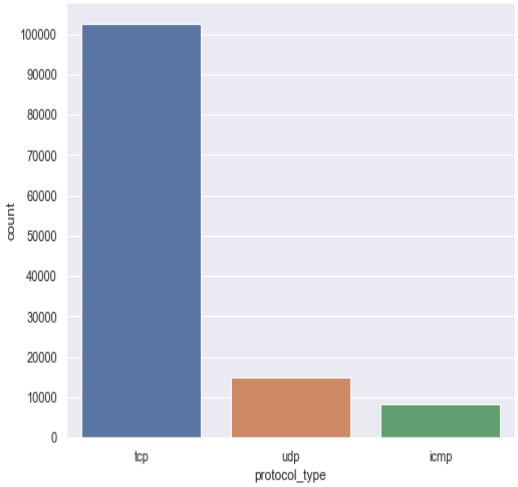

Figure 5. Protocol type visualization

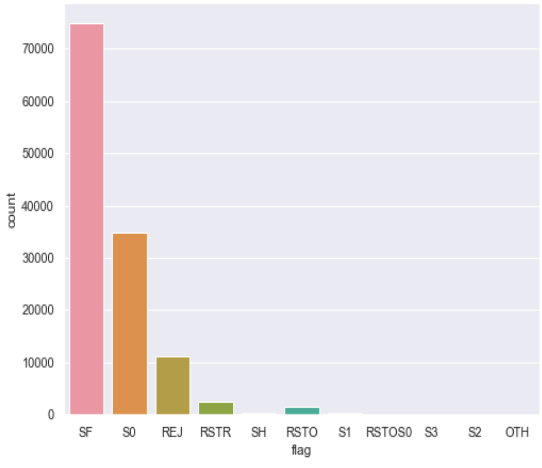

Figure 6. Flag visualization

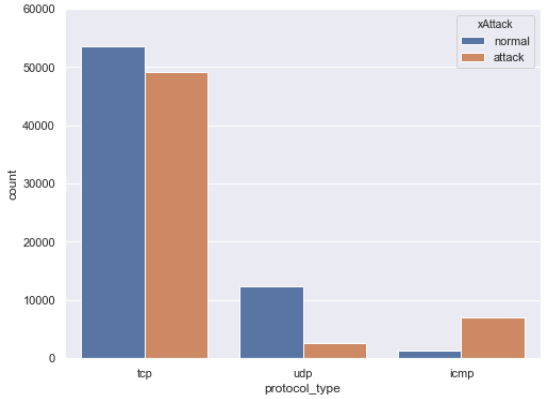

Figure 8. protocol type (Normal & Attack)

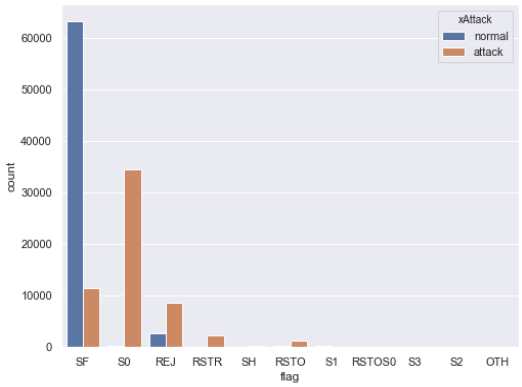

Figure 7. Flag distribution (normal &attack)

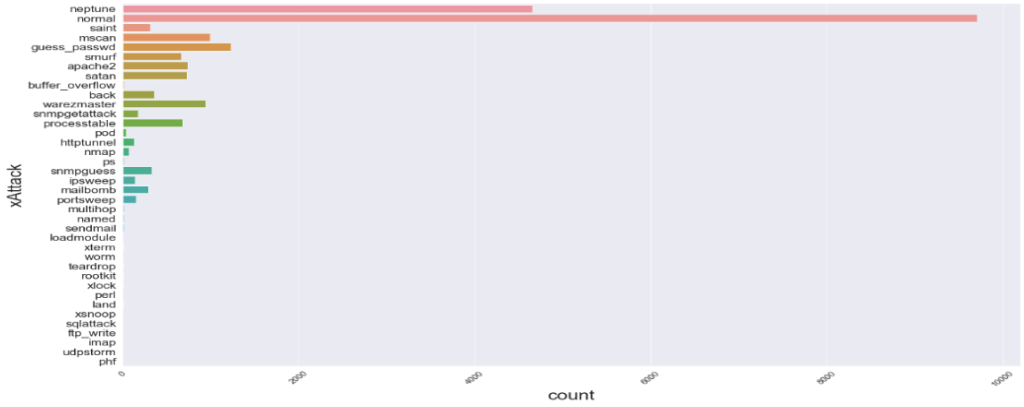

Figure 9. Visualization for XAttack column in train data

B) Data conversion & scaling

We display first five rows in train data As shown in Table (5) we notice that some features are numerical and the others categorical also the features have values differ a lot from each other so we have to convert all categorical features to numerical and scale down all data in the same range, the features will be scaled so that they will have the properties of a standard normal distribution with μ=0 and σ =1 to facilitate classification step and obtain precise results [23].

$$z = \frac{x-\mu}{\sigma} \qquad (4)$$

Where: z-score x: feature value, μ : mean, σ: standard deviation.

Table 5. Train Data Sample before Pre-processing

srv_diff_host_rate	dst_host_srv_diff_host_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	xAttack
0.00	0.00	1.00	1.00	attack
0.00	0.00	1.00	1.00	attack
0.00	0.02	0.00	0.00	normal
1.00	0.28	0.00	0.00	attack
0.75	0.02	0.83	0.71	attack

Table 6. Train Data Sample after Pre-processing.

srv_diff_host_rate	dst_host_srv_diff_host_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	xAttack
-0.386963	-0.229980	1.979791	1.929116	0
-0.386963	-0.229980	1.979791	1.929116	0
-0.386963	0.004234	-0.602719	-0.565483	1
3.557193	3.049016	-0.602719	-0.565483	0
2.571154	0.004234	1.540764	1.205682	0

Table (6) show that all data converted to numerical and are in the same range. We conducted the same steps in test data also to prepare it for validating.

C) Data Correlation

Correlation is considered a popular and effective technique for choosing the most related features in any dataset. It describes strength of association between features [22]. The following equation described the evaluation function

$$M_s = \frac{K\,\overline{r_{fc}}}{\sqrt{K+K(K-1)\overline{r_{ff}}}} \qquad (5)$$

Where, S is feature subset containing K features, $\overline{r_{fc}}$ is the mean feature-class correlation, $\overline{r_{ff}}$ is the average feature-feature correlation.

Figure (10) display correlation between all 16 features with each other and class.

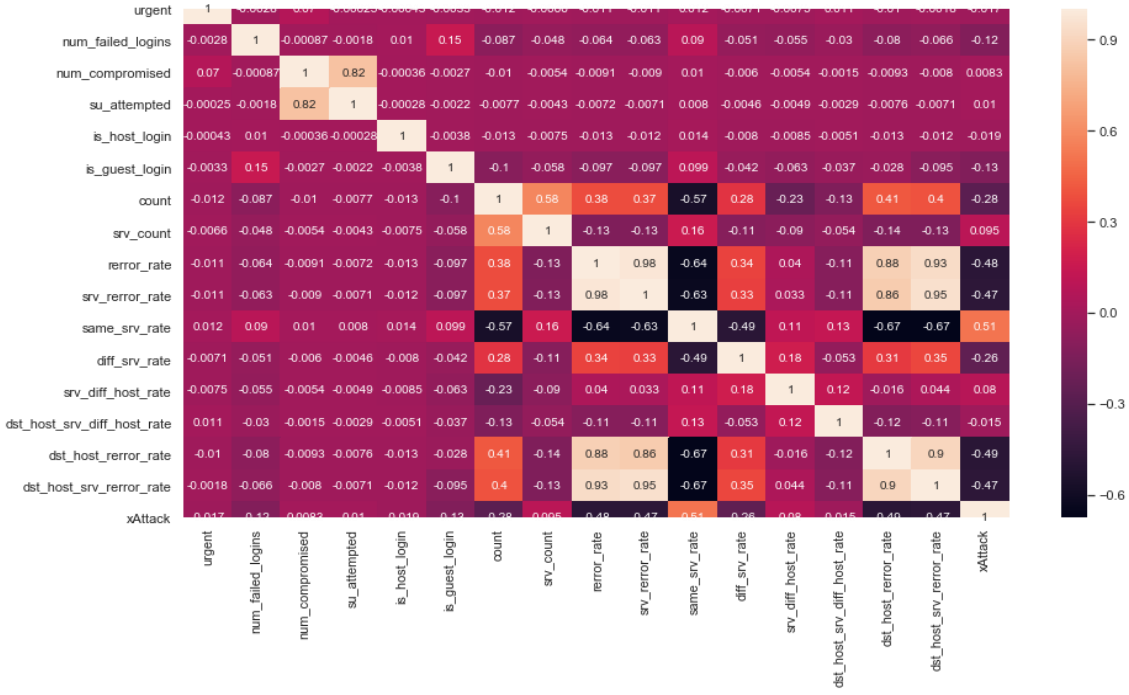

Figure 10. Correlation coefficients between all 16 attributes

8.2.2. Classification

We used for classification step many machine learning classifiers. Naïve Bayes classifier considered a supervised machine learning algorithm works on the principle of conditional probability as given by the Bayes theorem [28]. Support vector machine is one of the most popular supervised machine learning algorithms, which can efficiently perform linear and nonlinear classification1 [29]. Decision tree learning and Random forest are predictive modelling approaches used in statistics, data mining and machine learning [30],[31]. The k-nearest neighbour (k-NN) algorithm is a non-parametric method proposed for classification and regression by Thomas Cover [32].

8.2.3. Results

Our typical procedure is first training the model using a dataset, once it is built the next step is to use a dataset for testing your model and It is basically return the result, then those results will be compared with the truth to measure the accuracy of the model. They are several ways to perform these steps, the first way:

Use all available dataset for training and test on different dataset that means feed your data to the model and test with different one. The second way is split available data set in to training and test sets. The last way is Cross validation that means a technique involves reserving a particular sample of dataset on which you don't train the model, later you test the model on this sample before finalizing the model. Because of having a different kind of test and training data, we get good accuracy and the model will be able to deliver very good results.

In the 1st experiment we used for training (train data) and a different dataset for testing (test-data). All pre-processing steps also conducted into the test data with same steps in our three subsets. The results are discussed in the following table.

Table 7. Difference between three subsets with machine learning algorithms

Selected Features	**1st subset No. of features (39)**			**2nd subset No. of features (16)**			**3rd subset No. of features (4)**		
Performance Metrics / **Algorithms**	**Precision (%)**	**Recall (%)**	**F-score (%)**	**Precision (%)**	**Recall (%)**	**F-score (%)**	**Precision (%)**	**Recall (%)**	**F-score (%)**
NB	60.21	57.04	**41.88**	83.18	77.6	**77.37**	49.70	99.04	**65.70**
RF	66.38	93.75	**77.66**	91.27	92.2	**91.76**	64.30	74.00	**68.80**
SVM	74.05	95.85	**83.44**	73.46	94.0	**82.49**	62.95	89.72	**73.91**
DT	73.76	83.84	**78.74**	91.91	91.8	**91.86**	72.24	72.24	**73.42**
KNN	67.11	96.94	**79.31**	94.17	83.3	**88.39**	86.70	71.4	**72.71**

8.3. The Second Experiment

From previous experiment we found that [16-feature] {9,11,13,15,21,22,23,24,27,28,29,30,31,37,40,41} have a higher accuracy than other subsets, so it will be used in this experiment, we discovered that not all features content are important in attack detection. We named the feature content {Genes}. We will eliminate unimportant genes in every [16-feature]. As shown in table (8) and figure (10) we choose only one effective gene in every feature depending on its frequency with attacks.

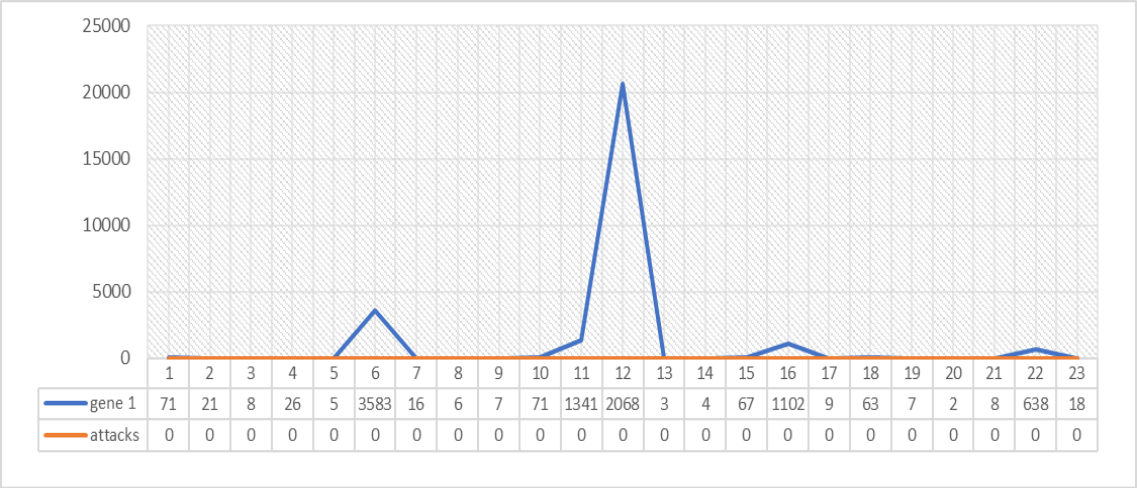

Figure 12. Gene 1 Distribution in feature 23

Table 8. Attacks symbols

A1	back	A9	multihop	A17	rootkit
A2	buffer_overflow	A10	neptune	A18	satan
A3	ftp_write	A11	nmap	A19	smurf
A4	guess_passwd	A12	normal	A20	spy
A5	imap	A13	perl	A21	teardrop
A6	ipsweep	A14	phf	A22	warezclient
A7	land	A15	pod	A23	warezmaster
A8	loadmodule	A16	portsweep		

8.3.1. Data Visualization

For example, feature number 23 which named (count) has 512 gene. We convert all 23 attack types to (A1,A2,…….A23) as shown in table (7), after obtaining frequency between these genes and attacks we found that only gene (1) in feature 23 has a higher attack detection. We will do that for all 16 features in our subset. The results are appeared in table (9). Figure (12) shows the distribution of gene 1 in feature23. Figures (11.13.14) display the Attacks behaviour with selected genes.

Table 9. Frequency between attacks and 16 features

Feature	Gene	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
9	0	956	30	6	53	11	3599	18	9	7	41214	1493	67337	3	4	201	2931	9	3633	2646	2	892	890	20
11	0	956	30	8	1	11	3599	18	9	7	41214	1493	67275	3	4	201	2931	9	3632	2646	2	892	890	20
13	0	73	12	7	53	10	3598	18	7	5	41214	1493	66971	3	4	201	2931	7	3630	2646	2	892	890	20
15	0	956	30	8	53	11	3599	18	9	7	41214	1493	67264	3	4	201	2931	10	3633	2646	1	892	890	20
21	0	956	30	8	53	11	3599	18	9	7	41214	1493	67342	3	4	201	2931	10	3633	2646	2	892	890	20
22	0	956	30	6	52	11	3599	18	9	5	41214	1493	66470	3	4	201	2931	10	3632	2646	2	892	584	18
23	1	71	21	8	26	5	3583	16	6	7	71	1341	20687	3	4	67	1102	9	63	7	2	8	638	18
24	1	47	24	7	26	3	1058	4	7	6	2462	407	16583	3	3	15	1648	9	2417	7	2	9	633	18
27	0	742	28	8	3	11	3172	17	9	7	34014	1493	64146	3	4	201	53	10	1493	2646	2	814	887	20
28	0	596	29	8	3	3	3173	18	9	7	34243	1493	63765	3	3	201	98	10	1657	2646	2	892	888	20
29	1	947	27	8	53	11	3590	16	6	7	440	1476	63373	3	4	201	2126	10	281	2646	2	682	883	20
30	0	947	27	8	53	11	3591	16	6	7	440	1477	62773	3	4	201	2128	10	282	2646	2	682	883	20
31	0	743	30	7	53	3	1066	4	9	6	40992	542	43019	3	3	99	2925	10	3624	2646	2	892	876	20
37	0	956	23	4	52	11	194	4	5	7	40801	535	33755	3	4	85	2913	9	3614	2646	2	892	369	20
40	0	54	25	8	2	10	3111	16	9	5	34157	1493	60001	2	4	181	3	9	645	2169	2	520	734	18
41	0	54	25	8	2	11	3159	18	8	7	34327	1493	61094	3	4	201	96	9	1657	2646	2	892	880	20

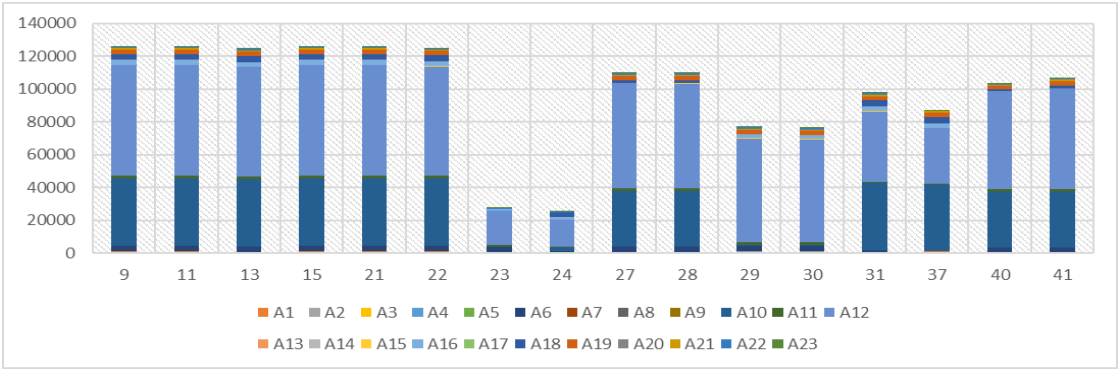

Figure 13. Distribution of 23 attacks in our feature set (16 features)

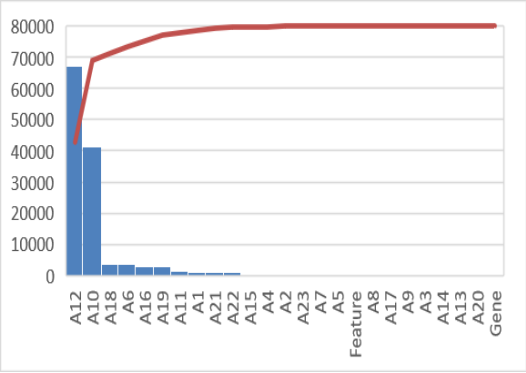

Figure 11. attack behaviour with selected genes

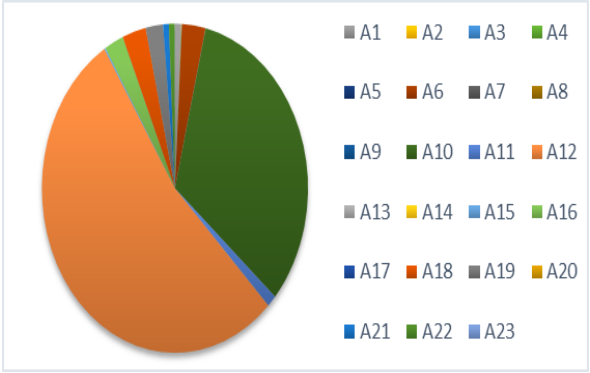

Figure 14. Distribution of attacks

8.3.2. Results

In the 2nd experiment we feed our model with training set and for testing, we split the training data into 70 % for training and 30 % for testing to measure the accuracy in a good way. We used same algorithms in classification as 1st experiment. The results show that selected genes have

accurate results than those in all 16-features. That is confirmation that not all feature content is important in attack detection as shown in table 10.

Table 10. Difference between all 16-feature and selected genes

Performance Metrics	(All 16 features)			All 16 features (selected genes)		
Algorithms	**Precision**	**Recall**	**F1_score**	**Precision**	**Recall**	**F1_score**
Random Forest	88.5 %	90.9 %	**89.68 %**	90.06 %	100 %	**94.76 %**
NB	83.18%	77.6%	**77.3 %**	82.72%	90.95%	**86.6%**
KNN	91.11 %	83.81 %	**87.30 %**	90.06 %	100 %	**94.76 %**
SVM	72.56 %	95.48 %	**82.45 %**	90.06 %	100 %	**94.76 %**
Decision Tree	88.73 %	90.01 %	**89.18 %**	90.06 %	100 %	**94.76 %**

8.4. The Third Experiment

In this experiment deep learning has been used through classification step by multi-layer perceptron (MLP). We performed this experiment with our three subsets (39-Feature, 16-Feature, 4-Features) and differentiate between the results. MLP used for training a supervised learning method called back-propagation. MLP is distinguished from a linear perceptron by several layers and non-linear activation. Data that cannot be linearly separated can be distinguished by MLP [33]. A total number of instances in all subsets is 125973 instances.

8.4.1. Performance metrics

To evaluate the experiment performance, seven known statistical indices (as mostly used in academic studies) are used to help rank the output of the classification. The correctly classified instances mean the sum of TP and TN. Similarly, incorrectly classified instances mean the sum of FP and FN. The total number of correctly instances divided by a total number of instances gives the accuracy. The Kappa statistics detect how closely the machine-learning classification instances matched the data labelled as the basic truth to test for the accuracy of a random classifier determined by the predicted accuracy. This implies that a value greater than 0 is better for your classifier. The Mean absolute error (MAE) the amount of predictions used to measure the possible result. The Root mean square error (RMSE) measure values difference model of an estimator predict and the values observed. The relative squared error (RSE) & The root relative squared error (RRSE) Is relative to what a simple predictor would have been if used. This basic indicator is more precisely just the sum of the actual values.

8.4.2. Results

Waikato Environment For knowledge Analysis (Weka) version 3.8.3 have been used as a simulation tool by OS windows 10 enterprise Intel® Core™ i5-3230M CPU@ 2.60GH, (RAM) 6.00 GB. We used for testing “Cross-Validation” with 5 folds, the.dataset.is divided into five parts of approximately. the. same size [34]. Tables (A, B, C) describe the MLP performance with 23 type of attacks which NSL-KDD dataset contains, by our three subsets (39, 16, 4 Features) in this order. The differences between our three subsets and the result of the third experiment are shown in table (11). Comparative discussion represented in table (12 &13) and Figure (17,18).

9. RESULTS DISCUSSION AND PERFORMANCE ANALYSIS

From our three experiment we have best and worst way. If we used all dataset features without feature selection techniques, we prefer using SVM algorithm which had a higher accuracy than other machine learning algorithms with (83.44%). If we use all dataset features with deep learning algorithm (MLP), we got better accuracy than using machine learning algorithms with (94.5%). If we used [16-feature] subset after applying features reduction techniques, we prefer using machine learning algorithms (RF) with accuracy (91.76%) and (DT) with accuracy (91.86%). When we selected genes from [16-feature] subset we get high accuracy than using all feature content with (94.76%) using machine learning algorithms. If we use [16-feature] subset with deep learning approaches, we got a higher accuracy (98.81%) with (MLP) technique. If we use [4-feature] subset with (MLP) deep learning technique we got higher accuracy (88.7%) than using machine learning algorithms with accuracy (73.9%). We notice that [16-feature] subset which has been chosen based on frequency between features and attacks has the high results in three experiments. Finally, we have bad precision if we use all the features in our dataset without feature selection. If we used 4-features also get bad accuracy because we do not make any feature selection methods, we just get it as a common feature from previous researches. When using selection methods with 16-attributes, we are more precise. Feature selection is considered a very important step before classification. Using Multi-layer perceptron deep learning technique got higher accuracy in all three experiments with all subsets.

Table 11. The performance of MLP technique in attack detection

Performance metrics	Time taken Build model	Correctly classified instances	Incorrectly Classified instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
1st subset 39-features	16731.4 Sec.	119057 **94.50 %**	6916 5.49 %	**0.90**	0.006	0.0635	12.7 %	39.207 %
2nd subset 16-features	1814.9 Sec.	124478 **98.81 %**	1495 1.18 %	**0.98**	0.001	0.0303	2.60 %	18.7015 %
3rd subset 4-features	5539.4 Sec.	111856 **88.79 %**	14117 11.20 %	**0.80**	0.013	0.0819	24.9 %	50.507%

Table 12. Comparison of F-score between all subsets

Algorithms	Machine learning algorithms (F-Score)					Deep learning (MLP) (F-Score)
Subsets	NB	RF	SVM	DT	KNN	
39-Features	41.8%	77.6%	83.4%	78.7%	79.3%	94.5%
16-Features	*77.3%*	*91.7%*	*82.4%*	*91.8%*	*88.3%*	*98.8%*
4- Features	65.7%	68.8%	73.9%	73.4%	72.7%	88.7%

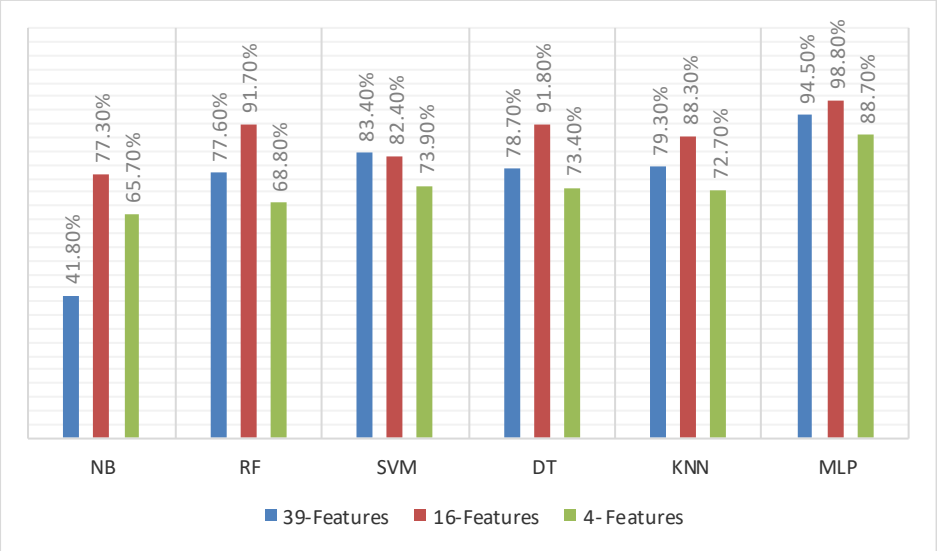

Figure16. (16-features) F score with machine and deep learning

Table 13. Comparison of F-score between 16 features (before & after deleted genes)

Measurement metric	F-Score				
Algorithms	NB	RF	SVM	DT	KNN
All 16-featrues	77.3%	89.6%	82.4%	89.1%	87.3%
All 16-featrues (Selected Genes)	*86.6%*	*94.7%*	*94.7%*	*94.7%*	*94.7%*

Figure 17. (16-features before & after deleted genes)

10. CONCLUSION

Machine and Deep Learning algorithms have been used in this paper to improve the classifications of intrusion detection systems. We applied three experiments using NSLKDD dataset. This dataset is divided into three subsets using featuring reduction approaches. In the 1st experiment we used NB, SVM, KNN, RF and DT algorithms for classification we noticed that the 16-feature subset had the highest results. We proposed in the 2nd experiment a model established on selected genes from every feature. when we eliminate unimportant genes from each feature, we will obtain a higher accuracy than using the all feature content. In the 3rd experiment we used Deep Learning Technique with Multilayer Perceptron (MLP) for classification. From these experiments we found that subset which has 16-feature has high accuracy and not all feature contents are important for attack detection. Our future work is to focus research on anew datasets as UNSW-NB-15 which contains up-to-date attacks. Using NS3 or opnet we can try our system in life attack scenario.

Table (A)

F-Measure	ROC Area	PRC Area	Class
0.991	0.996	0.993	normal
1.000	1.000	1.000	neptune
0.715	0.926	0.667	warezclient
0.972	1.000	0.991	ipsweep
0.994	0.999	0.996	portsweep
0.998	1.000	0.999	teardrop
0.952	0.996	0.959	nmap
0.977	0.994	0.979	satan
0.989	1.000	0.994	smurf
0.899	0.993	0.870	pod
0.727	0.960	0.713	back
0.831	0.998	0.931	guess_passwd
0.699	0.633	0.734	ftp_write
0.891	0.816	0.004	multihop
0.698	0.639	0.554	rootkit
0.911	0.881	0.187	buffer_overflow
0.794	0.871	0.110	imap
0.207	0.787	0.111	warezmaster
0.436	0.540	0.235	phf
0.200	0.856	0.226	land
0.365	0.551	0.254	loadmodule
0.347	0.636	0.123	spy
0.258	0.819	0.125	perl

Table (B)

F-Measure	ROC Area	PRC Area	Class
0.954	0.981	0.976	normal
0.982	0.995	0.993	neptune
0.305	0.935	0.413	warezclient
0.903	0.961	0.892	ipsweep
0.911	0.984	0.903	portsweep
0.571	0.910	0.595	teardrop
0.762	0.971	0.698	nmap
0.889	0.968	0.879	satan
0.779	0.955	0.806	smurf
0.372	0.681	0.267	pod
0.615	0.976	0.630	back
0.931	0.974	0.892	guess_passwd
0.699	0.341	0.734	ftp_write
0.891	0.660	0.004	multihop
0.698	0.762	0.124	rootkit
0.911	0.777	0.003	buffer_overflow
0.794	0.704	0.187	imap
0.207	0.829	0.001	warezmaster
0.436	0.690	0.235	phf
0.200	0.482	0.032	land
0.365	0.570	0.254	loadmodule
0.347	0.498	0.123	spy
0.258	0.669	0.125	perl

Table(C)

F-Measure	ROC Area	PRC Area	Class
0.954	0.983	0.981	**normal**
0.923	0.993	0.984	**neptune**
0.715	0.982	0.219	**warezclient**
0.756	0.982	0.635	**ipsweep**
0.054	0.941	0.234	**portsweep**
0.998	0.968	0.177	**teardrop**
0.952	0.941	0.181	**nmap**
0.251	0.945	0.280	**satan**
0.925	0.998	0.876	**smurf**
0.899	0.986	0.084	**pod**
0.727	0.844	0.027	**back**
0.036	0.994	0.194	**guess_passwd**
0.699	0.850	0.734	**ftp_write**
0.891	0.749	0.004	**multihop**
0.698	0.823	0.001	**rootkit**
0.911	0.938	0.017	**buffer_overflow**
0.794	0.465	0.110	**imap**
0.207	0.942	0.016	**warezmaster**
0.436	0.527	0.235	**phf**
0.200	0.199	0.226	**land**
0.365	0.692	0.254	**loadmodule**
0.347	0.643	0.123	**spy**
0.258	0.495	0.125	**perl**

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES:

[1] Zargar, G. R. “Category Based Intrusion Detection Using PCA”. International Journal of Information Security, 3, 259-271, October 2012.

[2] Das, A., Nguyen, D., Zambreno, J., Memik, G. and Choudhary, A. “An FPGA-Based Network Intrusion Detection Architecture, IEEE Transactions on Information Forensics and Security”, Vol. 3, No. 1, pp. 118-132, 2008.

[3] H.J. Liao, C.H.R. Lin, Y.C. Lin and K.Y. Tung, "Intrusion detection system: A comprehensive review.", Journal of Network and Computer Applications, Vol.36, issue.1, pp. 16-24, 2013.

[4] Chou, T. S. Yen, K. K. and Luo, J. Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms, International Journal of Computational Intelligence, Vol. 4, No. 3, pp. 196-208, 2008.

[5] S. Revathi, and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection." International Journal of Engineering Research & Technology (IJERT) 2, no. 12, 1848-1853, 2013.

[6] A. Alazab, M. Hobbs, J. Abawajy and M. Alazab. "Using feature selection for intrusion detection system." In 2012 international symposium on communications and information technologies (ISCIT), pp. 296-301. IEEE, 2012.

[7] L.Dhanabal and S.P. Shantharajah, " A study on NSL-KDD dataset for intrusion detection system based on classification algorithms", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, pp.446-452, June 2015

[8] M. Tavallaee, N. Stakhanova and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol.40, issue 5, pp. 516-524 ,2010.

[9] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE. 2009.

[10] A. Shrivastava, J.Sondhi and S. Ahirwar, "Cyber attack detection and classification based on machine learning technique using nsl kdd dataset", Int. Reserach J. Eng. Appl. Sci., vol.5, issue2, pp.28-31, 2017.

[11] D. H. Deshmukh, T. Ghorpade and P. Padiya, "Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset", 2015 International Conference on Communication, Information & Computing Technology (ICCICT) (pp. 1-6). IEEE, 2015.

[12] B. Ingre and A. Yadav," Performance analysis of NSL-KDD dataset using ANN", 2015 international conference on signal processing and communication engineering systems (pp. 92-96). IEEE. 2015.

[13] H.Chae and S.Choi," Feature Selection for efficient Intrusion Detection using Attribute Ratio", INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS, Vol. 8, pp.134-139, 2014.

[14] M.Ambusaidi, X.He, P.Nanda and Z.Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," IEEE Transactions on Computers, vol. 65, no.10, pp. 2986-2998, 2016.

[15] Y.Wahba, E.ElSalamouny, and G. ElTaweel, "Improving the performance of multi-class intrusion detection systems using feature reduction" , IJCSI International Journal of Computer Science Issues , Vol. 12, Issue 3, pp.255-262, May 2015.

[16] D.A.Kumar and S.R.Venugopalan, "The effect of normalization on intrusion detection classifiers (Naïve Bayes and J48)", International Journal on Future Revolution in Computer Science & Communication Engineering ,Vol 3, Issue: 7, pp.60-64, July 2017.

[17] A.Puri and N.Sharma, "A NOVEL TECHNIQUE FOR INTRUSION DETECTION SYSTEM FOR NETWORK SECURITY USING HYBRID SVM-CART." International Journal of Engineering Development and Research, Vol. 5, Issue.2, pp.155-161, 2017.

[18] M. Abdullah, A. Alshannaq, A. Balamash and S. Almabdy. "Enhanced intrusion detection system using feature selection method and ensemble learning algorithms" International Journal of Computer Science and Information Security (IJCSIS), Vol. 16, No. 2, pp.48-55, 2018.

[19] L. Gnanaprasanambikai and N.Munusamy, "Data Pre-Processing and Classification for Traffic Anomaly Intrusion Detection Using NSLKDD Dataset. Cybernetics and Information Technologies", CYBERNETICS AND INFORMATION TECHNOLOGIES Vol.18, pp.111-119, 2018.

[20] K. A.Taher, B. M. Y .Jisan and M. M. Rahman,"Network intrusion detection using supervised machine learning technique with feature selection", 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 643-646, 2019

[21] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model", Journal of Computational Science, 2018 - Elsevier, vol.25, pp.152-160, 2018.

[22] M. A. Hall, "Correlation-based feature selection for machine learning", 1999.

[23] Patro, S., and Kishore Kumar Sahu. "Normalization: A preprocessing stage." arXiv preprint arXiv:1503.06462 (2015)

[24] A. Alazab, M. Hobbs, J. Abawajy and M. Alazab. "Using feature selection for intrusion detection system." In 2012 international symposium on communications and information technologies (ISCIT), pp. 296-301. IEEE, 2012.

[25] Tesfahun, Abebe, and D. Lalitha Bhaskari. "Effective hybrid intrusion detection system: A layered approach." International Journal of Computer Network and Information Security 7, no. 3 (2015): 35.

[26] Harb, Hany M., Afaf A. Zaghrot, Mohamed A. Gomaa, and Abeer S. Desuky. "Selecting optimal subset of features for intrusion detection systems." (2011).

[27] Latah, Majd, and Levent Toker. "Towards an efficient anomaly-based intrusion detection for software-defined networks." IET networks 7, no. 6 (2018): 453-459

[28] Maron, Melvin Earl. "Automatic indexing: an experimental inquiry." Journal of the ACM (JACM) 8, no. 3 (1961): 404-417.
[29] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.
[30] Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan et al. "Top 10 algorithms in data mining." Knowledge and information systems 14, no. 1 (2008): 1-37.
[31] Piryonesi, S. Madeh, and Tamer E. El-Diraby. "Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index." Journal of Infrastructure Systems 26, no. 1 (2020): 04019036.
[32] Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." The American Statistician 46, no. 3 (1992): 175-185.
[33] Rosenblatt, Frank. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. No. VG-1196-G-8. Cornell Aeronautical Lab Inc Buffalo NY, 1961.
[34] Stone, Mervyn. "Cross-validatory choice and assessment of statistical predictions." Journal of the Royal Statistical Society: Series B (Methodological) 36, no. 2 (1974): 111-133.
[35] Tavallaee, et al. “A Detailed Analysis of the KDD CUP 99 Data Set.” 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.
[36] Huffman, David A. "A method for the construction of minimum-redundancy codes." Proceedings of the IRE 40, no. 9 (1952): 1098-1101.

AUTHORS

HAJAR M. ELKASSABI is a researcher at Faculty of Engineering, Mansoura University, Egypt. She Received Th B.S.C. degree in Comm. Engineering from Mansoura University, Egypt in 2009. She received Diploma degree in electrical engineering from Mansoura University Egypt in 2012. She received Diploma degree from Ministry of Communication and Information Technology (MCIT) in 2014 track networking. Her current research interests include network security, Cloud Computing, and Internet routing.

MOHAMMED M. ASHOUR is an assistant professor at the faculty of engineering Mansoura University, Egypt. He received B.Sc. from Mansoura University Egypt in 1993. He received an M.Sc. degree from Mansoura University, Egypt in 1996. He receives a Ph.D. degree from Mansoura University, Egypt 2005. Worked as Lecturer Assistant at Mansoura University, Egypt from 1997, from 2005, an Assistant Professor. Fields of interest: Network Modelling and Security, Wireless Communication, and Digital Signal Processing.

FAYEZ W. ZAKI is a professor at the Faculty of Engineering, Mansoura University. He received the B. Sc. in Communication Eng. from Menofia University Egypt 1969, M. Sc. Communication Eng. from Helwan University Egypt 1975, and Ph.D. from Liverpool University 1982. Worked as a demonstrator at Mansoura University, Egypt from 1969, Lecture assistant from 1975, a lecturer from 1982, Associate Prof. from 1988, and Prof. from 1994. Head of Electronics and Communication Engineering Department Faculty of Engineering, Mansoura University from 2002 till 2005. He supervised several MSc and Ph.D. thesis. He has published several papers in refereed journals and international conferences. He is now a member of the professorship promotion committee in Egypt.