Egyptian Informatics Journal (2014) 15, 13–24

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

A proposed HTTP service based IDS

Mohamed M. Abd-Eldayem *

IT Department, Faculty of Computers and Information, Cairo University, Egypt
CEN Department, College of Computers and Information Sciences, King Saud University, Saudi Arabia

Received 9 October 2013; revised 1 January 2014; accepted 6 January 2014
Available online 30 January 2014

KEYWORDS

Computer security;
Network security;
Intrusion Detection System (IDS);
Naïve Bayes classifier

Abstract The tremendous growth of the web-based applications has increased information security vulnerabilities over the Internet. Security administrators use Intrusion-Detection System (IDS) to monitor network traffic and host activities to detect attacks against hosts and network resources. In this paper IDS based on Naïve Bayes classifier is analyzed. The main objective is to enhance IDS performance through preparing the training data set allowing to detect malicious connections that exploit the http service. Results of application are demonstrated and discussed. In the training phase of the proposed IDS, at first a feature selection technique based on Naïve Bayes classifier is used, this technique identifies the most important HTTP traffic features that can be used to detect HTTP attacks. In the testing and running phases proposed IDS classifies the network traffic based on the requested service, then based on the selected features Naïve Bayes classifier is used to analyze the HTTP service based traffic and identifies the HTTP normal connections and attacks. The performance of the IDS is measured through experiments using NSL-KDD data set. The results show that the detection rate of the IDS is about 99%, the false-positive rate is about 1%, and the false-negative rate is about 0.25%; therefore, proposed IDS holds the highest detection rate and the lowest false alarm compared with other leading IDS. In addition, the proposed IDS based on Naïve Bayes is used to classify network connections as a normal or attack. And it holds a high detection rate and a low false alarm.

1. Introduction

These days, most of the universities, organizations and companies provide their services through a computer network and Internet technologies; therefore, most of their tasks are accomplished using web-based applications. However, the attackers could have exploited the Internet to break into the local network to gain the confidential information or to compromise the network resources. Some security tools such as firewalls, anti-virus software and Intrusion-Detection Systems (IDSs) are used to protect the network and to thwart hackers. IDSs are used to monitor network traffic to detect the intruder

* Address: CEN Department, College of Computers and Information Sciences, King Saud University, Saudi Arabia. Tel.: +966 592626083; fax: +966 014676990.
E-mail address: mdayem@gmail.com

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

1110-8665 © 2014 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.
http://dx.doi.org/10.1016/j.eij.2014.01.001

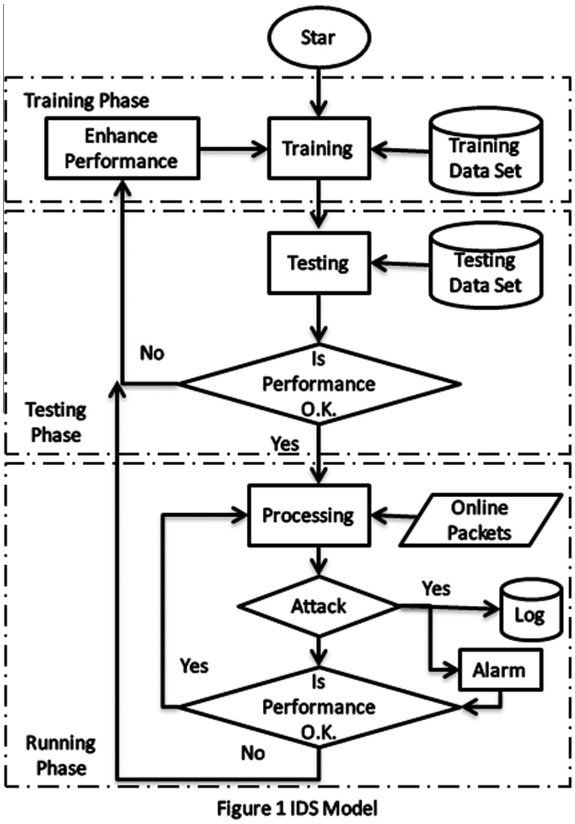

Figure 1 IDS model.

network events. Statistical methods are utilized widely in the IDSs since these methods are analytical methods that depend on the probabilistic, and they are considered as the major tools that are coping with uncertain data. Many of these methods are used to enhance the IDS performance; they give a better way to classify network events as normal or as attacks. Bayesian classifier is an example of the statistical methods that are used in IDSs. In the training phase, it computes the conditional probabilities for each of the normal and the attack classes. It uses a training data set that is classified into attack and normal classes. And in the testing and running phase the classifier uses these probabilities to extract the belongings probabilities of different classes; therefore, the unknown network traffic can be classified as the class that supposes maximum value [1,2]. This means that the running network traffic would be classified as normal or as attack events. In this paper, IDS based on Naïve Bayes classifier is proposed and its performance is tested through practical experiments. The following sections of this paper are organized as follows: the second section summarizes the ideas of the leading IDSs. Section 3 describes the proposed IDSs. The implementations, discussions and experimental results for these IDSs are described and illustrated in Section 4, finally the conclusions and future works are summarized in Section 5.

2. Related works

In [3] anomaly detection IDS is proposed, it combines k-Means, K-nearest neighbor classifier and Naive Bayes classifier. It selects the important features using an entropy based algorithm, then it applies k-Means in clusters' phase. It is not only able to detect the attacks, but also it can classify them into four types: DOS, U2R, R2L and probe. This system can achieve 98.18% detection rate and 0.83% false-positive rate; however, it may increase the processing time because it executes the k-Means, K-nearest neighbor classifier and Naive Bayes classifier. In addition, the accuracy of classifying the attack types ranges from 92% to 98%. A score-based multi-cycle detection algorithm based on Shiryaev–Roberts procedure is proposed in [4]. Comparing to similar procedures; Shiryaev–Roberts procedure is computationally inexpensive in addition it is easy to be practically applied in real time IDS. The target

Table 1 List of features of a record in KDD data set.

Index	Feature	Index	Feature
1	duration	22	is_guest_login
2	protocol_Type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src-bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_error_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_error_rate
21	is_hot_login		

Table 2 Lookup table of feature 2.

Feature 2	Index
icmp	1
tcp	2
udp	3

Table 3 Lookup table of feature 3.

Feature 3	Index	Feature 3	Index	Feature 3	Index
aol	1	http_8001	25	red_i	49
auth	2	imap4	26	remote_job	50
bgp	3	IRC	27	rje	51
courier	4	iso_tsap	28	shell	52
csnet_ns	5	klogin	29	smtp	53
ctf	6	kshell	30	sql_net	54
daytime	7	ldap	31	ssh	55
discard	8	link	32	sunrpc	56
domain	9	login	33	supdup	57
domain_u	10	mtp	34	systat	58
echo	11	name	35	telnet	59
eco_i	12	netbios_dgm	36	tftp_u	60
ecr_i	13	netbios_ns	37	tim_i	61
efs	14	netbios_ssn	38	time	62
exec	15	netstat	39	urh_i	63
finger	16	nnsp	40	urp_i	64
ftp	17	nntp	41	uucp	65
ftp_data	18	ntp_u	42	uucp_path	66
gopher	19	other	43	vmnet	67
harvest	20	pm_dump	44	whois	68
hostnames	21	pop_2	45	X11	69
http	22	pop_3	46	Z39_50	70
http_2784	23	printer	47		
http_443	24	private	48		

Table 4 Lookup table of feature 4.

Feature 4	Index
OTH	1
REJ	2
RSTO	3
RSTOS0	4
RSTR	5
S0	6
S1	7
S2	8
S3	9
SF	10
SH	11

Table 5 Results of the Naive-Bayes classifier for 30 features.

	Attack	Normal
Attack	TP = 63.34%	FN = 36.66%
Normal	FN = 6.48%	TN = 93.52%

Table 6 Results of the Naive-Bayes classifier for 27 features.

	Attack (%)	Normal (%)
Attack	63.04	36.96
Normal	6.64	93.36

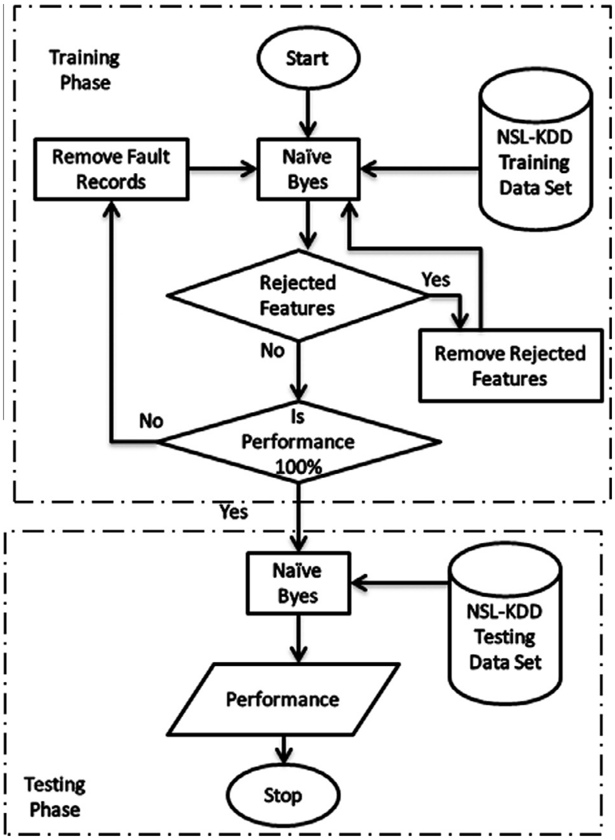

Figure 2 The first proposed IDS.

of this proposed ID is to identify the attacks as soon as it happened to minimize the detection delays; however, it increases the false alarm rates; therefore, it requires additional filtering technique with high detection accuracy, which it may increase the processing delay. Tree based IDS classification algorithms are evaluated in [5]. Comparing with other IDS; experimental results show that the Random Tree model could achieve detection accuracy equals 97.49%, and its false detection rate is about 2.5%. However, these accuracy and detection rate are still needed to be enhanced.

IDS based on integration of Snort and Bayes theorem is proposed to be implemented in the cloud environment [6]. In this IDS, the packets are captured from the network, then Snort checks these captured packets using the signature based detection technique; suspicious packets are stored within the log database; other packets are checked again, based on the behavior Bayesian classifier identifies intrusion packets and normal packets. Intrusion packets are logged in the alert database while normal packets are allowed in the system as legitimate packets. This IDS achieves 96% detection rate and 1.5% false positive rate. As it uses two successive filtering techniques, it may cause processing overhead, which increases delay, however, in my opinion, this delay can be reduced by parallel execution of both filtering techniques using parallel programming.

Naive Bayes (NB) classifiers are used in anomaly based IDS [7]; it is proposed to detect new intrusions to the Internet using the Routing Information Base (RIB) of the Border Gateway Protocol (BGP). Various feature selection algorithms are tested to be used to train the classifiers. These algorithms are: Fisher [8,9], minimum redundancy maximum relevance [10], extended/weighted/multi-class odds ratio and class discriminating measure [11]. The experimental results show that the weighted odds' ratio algorithm is the best feature selection algorithm based on F-score. However, the NB classifiers achieve the best detection accuracy (68.7%) if they are trained based on the multi-class odds ratio feature selection algorithm. In my opinion, the accuracy level of this IDS is low and this is because it depends on limited information (RIB) to select the features that are used to detect the new intrusions. In [12] IDS based on data mining techniques is proposed, it consists of four main modules: k-means clustering, Neuro-fuzzy training, SVM training vector and radial SVM classification modules. K-means clustering technique is used to group input data set into five clusters; these clusters are: one cluster belongs to normal behavior, and the other four clusters belong to four intrusions types. A Neuro-fuzzy classifier is associated for every cluster, and it is trained with the data of the respective cluster. SVM training vector is used to decrease the number of input data set attributes. This task simplifies the classification process and makes it more efficient. Then the data set with reduced attributes is used in radial SVM to detect intrusions. The accuracy of this technique is about 97.5% that are a relatively good accuracy level. The delay can be reduced if the clusters are processed in parallel programming.

3. The proposed IDSs

The IDS model is described in Fig. 1; It consists of three phases: Training, Testing and Running phase. The Training phase is an offline phase; it uses the training data set to train the IDS to identify the normal connections and to detect the attacked connections, the training data set must include adequate information for both normal connections and attacks. The Testing phase is offline phase, during this phase the testing data set that includes normal connections and attacks is fed into the IDS; the performance of the IDS is measured; the high IDS performance indicates high accuracy in detecting the normal connections and attacks. If the level of IDS performance is not accepted, then the IDS must be enhanced, and the training and testing phase are executed again. The sequence of IDS enhancement, training and testing is repeated until reaching the accepted IDS performance. After passing the testing phase the IDS can be used to protect real time network traffic, that is called Running phase. During this phase, the IDS may need to be enhanced to accurately detect the new types of attacks; therefore, the previous steps are repeated starting from the training phase.

3.1. Naive Bayes classifier

There are many recent IDSs researches exploits Bayesian theory (Eq. (1)) to classify network traffic as normal or as attack events [1,2].

$$P(C_j\ X) = \frac{P(X\ C_j)P(C_j)}{P(X)},$$

$$IFF\ P(C_j\ X) > P(C_i\ X), 1 \leq i \leq m, i \neq j \quad (1)$$

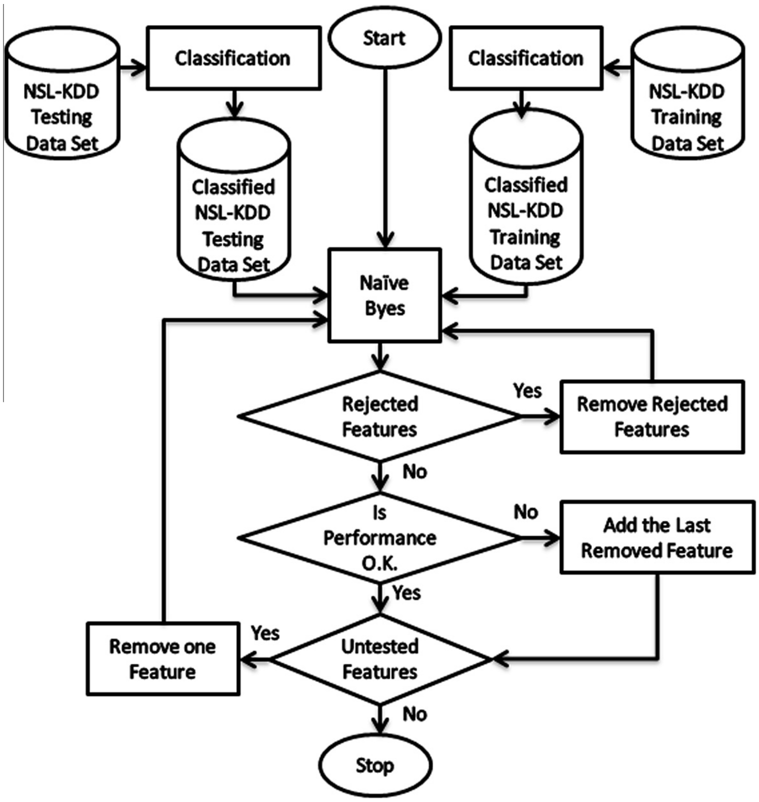

Figure 3 The second proposed IDS.

where Class $C_j \in$ Given set of m Classes $\{C_1, C_2, C_3, \ldots\ldots\ldots\ldots, C_m\}$, X is an unknown data sample and $P(X)$ is constant for each one category.

It is sufficient to determine only the numerator term because of $P(X)$ is constant for every 'X'; therefore, Naive Bayes classifier determines Eq. (2) to allocate a sample of unknown X to class C_j.

$$P(C_j\ X) = P(X\ C_j)P(C_j),$$
$$IFF\ P(C_j\ X) > P(C_i\ X), 1 \leq i \leq m, i \neq j \tag{2}$$

To apply a Naive Bayes classifier in IDS; Priori probability $P(C_j)$ can be determined using the training data set in Eq. (3), and if the sample has many attributes, then $P(C_j)$ can be determined using Eq. (4) [2].

$$P(C_j) = S_j/S \tag{3}$$

where S_j is the training sample size in the class C_j, and S is the total number of the training samples.

$$P(A\ C_j) = P(A_1\ C_j)P(A_2\ C_j)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots P(A_k\ C_j) \tag{4}$$

where A is the set of attributes $\{A_1, A_2, \ldots\ldots\ldots, A_k\}$, in the IDS A is the values of the set of features that characterize the network traffics.

Eq. (5) is used to classify records A in the test data set or in the online traffic.

$$Record\ A \in C_j$$
$$IFF\ P(A\ C_j)P(C_j) > P(A\ C_i)P(C_i), 1 \leq i \leq k, i \neq j \tag{5}$$

Eq. (5) assigns record A (one network connection) to class C_j if the probability of A to belong to C_j is the largest probability. The network connections could be classified as normal or as attack events, in this case, there are only two classes: class C_1 and C_2, and the IDS could classify the connection as normal or attack connection without identifying the type of attack. In addition, the classification process may produce several classes, one class for normal connections and one class for each attack type; therefore, the IDS could identify the type of attack.

3.2. The proposed IDS based on Naive Bayes classifier

The proposed IDSs use Naive Bayes equation (Eq. (5)). To classify network traffics as normal or as attacking connections based on their features. Many IDS researchers use the NSL-KDD data set [13] or KDD'99 data set [14] to evaluate their IDSs; data sets include training and testing data. NSL-KDD data set is an improved version of the KDD'99 data set, the NSL-KDD data set is simpler than KDD'99 data set, and it is easier to be used in Wintel platform; therefore, the NSL-KDD data set is used to train and to test the proposed IDS. Both of NSL-KDD and KDD'99 data sets include 41 features of the network connection; these features are listed in Table 1.

In the training phase, the IDS reads the NSL-KDD training data set, next it applies the Naive Bayes equation to classify the data records into the normal connection or attacks based on their features. Practically, some features are rejected because of they are not positive in class variance; therefore, they are removed from the data set. Naive Bayes classifier is trained another time using the reduced training data set, subsequently the performance of the classifier is measured using the training data set again. If there is any false alarm, subsequently the record causing this alarm are removed from the training data set and the IDS is retrained and the IDS performance is measured, these steps are repeated until the performance reaches 100%, after that the training phase is ended, and the IDS starts the testing phase. The target of the training phase is to select the records from the training data set that accurately classified by the probability distribution. This is to improve the accuracy of the IDS to identify the normal connections and attacks.

In the test phase, the NSL-KDD testing data set is used to measure the accuracy of the using Naive Bayes classifier. In Section 4the practical results show improvement to the performance of the Naive Bayes classifier using the proposed IDS.

3.3. The proposed HTTP service based IDS

The proposed HTTP based IDS is shown in Fig. 3, both training and testing data set are prepared through the classification process. In this process only HTTP traffic is selected; the resultant data sets consist of 38 features, features 2, 3 and 4 text-based features are removed, as shown in Table 1, these features are protocol, service and flag. All records of HTTP traffic had an identical protocol (TCP) and identical service (HTTP), in addition the attacked HTTP traffic is classified based on an attack type (for example: Neptune or Back attacks). The target of the classification process is creating the best probability distribution that describes each type of HTTP attack; the behavior of attacks is distinctive from one attack to another; therefore, each attack type may be described by specifying the probability distribution that is distinctive from the other attacks' probability distributions. The Naive Bayes classifier uses both testing and training data sets to classify the records of the traffic into the normal connection or attacks, and the types of attacks are identified. Practically, some features are rejected because they are not positive in class variance; therefore, they are removed from the data sets. The performance of the classifier is measured; one feature is removed from both training and testing data sets; the classifier performance is measured another time, if the performance is reduced, then the removed feature is added once again to the data sets, otherwise another feature is removed and its effect on performance is measured. The previous steps are repeated until the effects of all features on the classifier performance are checked. At the end, the proposed IDS identify the most important features that can be used to accurately detect HTTP attacks. As shown in Section 4, 13 features are the most important features that make the performance of the Naive Bayes classifier is about 99%.

4. The implementation and experimental results of the proposed IDS

The NSL-KDD data set [13] is used to measure the performance of the proposed IDSs. NSL_KDD data set includes 41 features of the network connection; these features are described in Table 1, and in this research, the performance of IDS based on Naive-Bayes is measured. A Matlab program is implemented to apply Naive-Bayes IDS using The NSL-KDD data set. All features must have a numeric value to be implemented in the Naive-Bayes classifier; however, the features 2,3 and 4 are attribute values; therefore, they must be converted to numeric values before execution of the IDS as shown in lookup Tables 2–4.

The Naive-Bayes classifier is implemented using KDD-Train as training file and KDD-Test as a testing file. During execution of the IDS, the error message indicates that features: 7, 8, 9, 11, 15, 16, 17, 18, 20, 21 and 22 are rejected because they are not positive in class variance; therefore, these features are removed and the number of features is reduced in 30. The experimental results (Confusion Matrix) are shown in Table 5.

As shown in Table 5, the performance of the IDS is 63.34% to detect attacks while it is 93.52% to detect normal traffic and the false-positive alarm are 36.66% and false-negative alarms are 6.48%. To measure the effect of the attribute features (2, 3 and 4) they are excluded from the KDD-test data; therefore, the number of features is reduced in 27. The experimental results' confusion matrices for 27 features are shown in Table 6.

As shown in Tables 5 and 6, there is no significant difference between the performance of the IDS before and after excluding the attribute features; therefore, in the following experiments, only 27 features are tested.

4.1. The First proposed IDS

The detection performance is low as shown in Table 6, and to improve it, it is proposed to achieve 100% detection rate during the training phase; it means that the training data must be adapted for creating a probability distribution that can be used to detect all intrusions in the training data before applying it to the testing data. As shown in Fig. 2, this target can be achieved by applying two steps: execute the IDS on the training data as training data and as testing data, identify the connections that cause false alarms to remove it from the training data, and these steps are repeated until no false alarm is generated.

Table 7 and Fig. 4 show the result of each step during execution of the first proposed IDS. Step 0 represents the results using the complete training data set KDD-Train 20% (25,192 records) as training and as testing data set. There are about 12.9% false-negative alarms and 9.6% false-positive alarms. The records that cause these alarms are removed from the data set; therefore, the data set will be reduced to include 22,328 records, then the test is repeated (step 1). After step 1, the detection rates are enhanced, the false-negative is 0.13%, and the false-positive is 99.64%. The test is repeated until the true-positive and true-negative rates reach 100% as shown in step 9. The training phase is finished, and the testing phase is started; the KDD-Train is used as training data set, and KDD-Test is used as the testing data set, the results are shown in step 10 in Table 7. The false-negative rate is about 24%, and the false-positive rate is 4.61%. As shown in Table 6 before applying the proposed modification, the false-negative rate equals about 37% while the false-positive rate is about 6.64%. Clearly, the false-negative and false-positive rates are

Table 7 The experimental results of the first proposed IDS.

Step	# of Attack records	# of Normal records	Total	True positive (%)	False negative (%)	True negative (%)	False positive (%)
0	11,743	13,449	25,192	90.37	12.89	87.11	9.63
1	10,612	11,716	22,328	99.87	0.13	99.64	0.36
2	10,533	11,529	22,062	99.88	0.12	99.87	0.13
3	10,520	11,514	22,034	99.91	0.09	99.95	0.05
4	10,511	11,508	22,019	99.93	0.07	99.92	0.08
5	10,504	11,499	22,003	99.91	0.09	99.97	0.03
6	10,495	11,495	21,990	99.94	0.06	99.97	0.03
7	10,489	11,491	21,980	99.98	0.02	100.00	0.00
8	10,487	11,491	21,978	99.99	0.01	100.00	0.00
9	10,486	11,491	21,977	100.00	0.00	100.00	0.00
10	12,833	9711	22,544	75.94	24.06	95.39	4.61

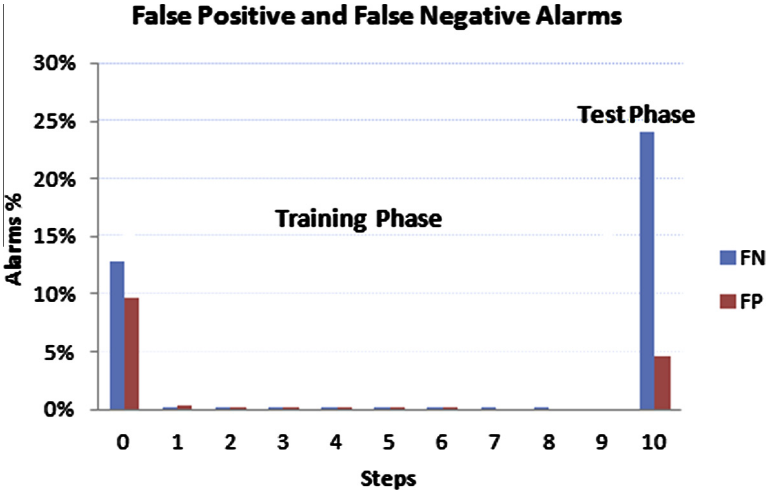

Figure 4 The experimental results of the first proposed IDS.

Table 8 HTTP based IDS experimental results.

Step	Features	Attack	Normal	Total	True positive (%)	False negative (%)	True negative (%)	False positive (%)	Comment or error message
0	38	1471	19,634	21,105	–	–	–	–	The within-class variance in feature 4, 5, 6, 8, 12 13, 14, 15, 17, 18 and 19 in class attack are not positive.
1	27	1471	19,634	21,105	100	0.00	98.25	1.75	After removing the above features
2	27	1471	19,290	20,761	100	0.00	98.24	1.78	
3	27	1471	18,951	20,422	–	–	–	–	The within-class variance in feature 15 and 16 in class normal are not positive.
4	25	1471	18,951	20,422	100	0.00	98.22	1.78	After removing the above features
5	25	1471	18,613	20,084	100	0.00	98.51	1.49	
6	25	1471	18,335	19,806	100	0.00	99.13	0.87	
7	25	1471	18,176	19,647	100	0.00	99.27	0.73	
8	25	1471	18,043	19,514	100	0.00	99.12	0.88	
9	25	1471	17,738	19,209	100	0.00	99.50	0.50	
10	25	1471	17,650	19,121	100	0.00	99.49	0.51	
11	25	1471	17,560	19,031	–	–	–	–	The within-class variance in feature 11 and 12 in class normal are not positive.
12	23	1471	17,560	19,031	100	0.00	99.38	0.62	After removing the above features
13	23	1471	17,452	18,923	100	0.00	99.89	0.11	
14	23	1471	17,433	18,904	–	–	–	–	The within-class variance in feature 8, 16 and 17 in class normal are not positive.
15	20	1471	17,433	18,904	99.52	0.48	99.83	0.17	After removing the above features
16	20	1464	17,404	18,868	100	0.00	99.91	0.09	
17	20	1464	17,388	18,852	100	0.00	99.80	0.20	
18	20	1464	17,340	18,804	100	0.00	99.92	0.08	
19	20	1464	17,338	18,802	100	0.00	99.99	0.01	
20	20	1464	17,338	18,802	100	0.00	100	0.00	End of training phase
21	20	6673	1180	7853	97.63	2.37	97.46	2.54	Testing phase

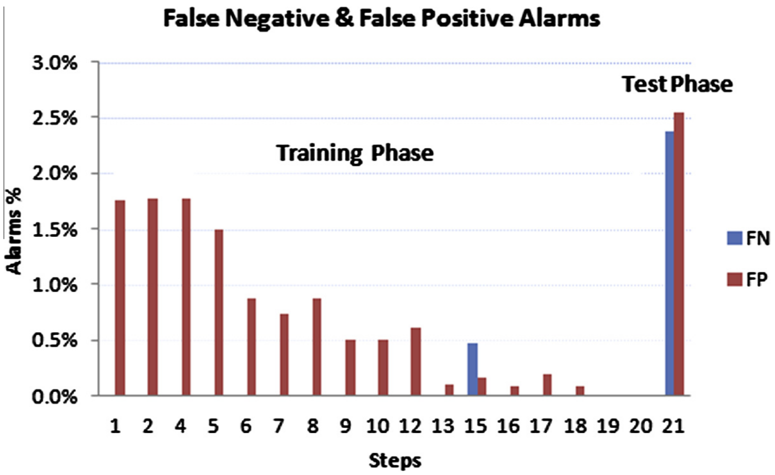

Figure 5 HTTP based IDS experimental results.

reduced. However, the proposed IDS enhances the detection rate, but more enhancement is required. In the following section, additional solution to enhance the IDS performance is proposed.

4.2. The second proposed IDS (HTTP based IDS)

To improve the IDS performance as described in Section 3 it is proposed to apply IDS to each service on separate. In this

Table 9 HTTP based IDS for 27 and 20 features.

Features	True positive (%)	False negative (%)	True negative (%)	False positive (%)	Comment
27	98.19	1.81	97.63	2.37	All data records
27	96.90	3.10	98.64	1.36	Without fault data records
20	99.04	0.96	95.68	4.32	All data records
20	97.63	2.37	97.46	2.54	Without fault data records

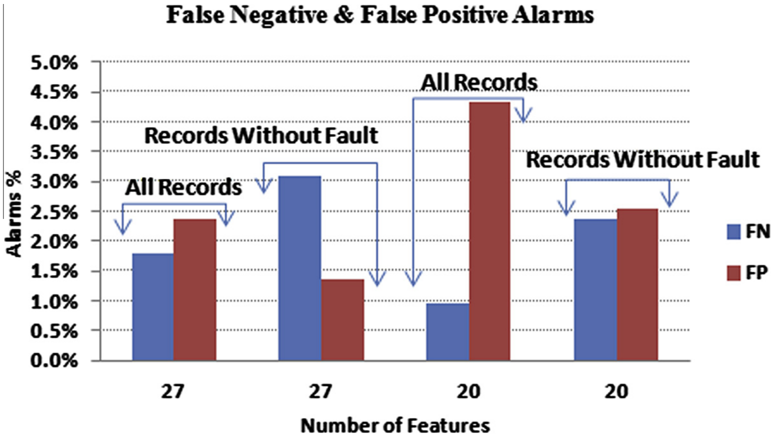

Figure 6 HTTP based IDS for 27 and 20 features.

Table 10 Experimental results HTTP based IDS for Neptune attack.

# of Features	True positive (%)	False negative (%)	True negative (%)	False positive (%)
Training phase 19 features	100.00	0.00	99.20	0.80
18	100.00	0.00	99.21	0.79
17	100.00	0.00	99.22	0.78
16	100.00	0.00	99.28	0.72
15	100.00	0.00	99.31	0.69
14	100.00	0.00	99.21	0.79
13	100.00	0.00	99.25	0.75
12	100.00	0.00	99.41	0.59
11	100.00	0.00	99.08	0.92
10	100.00	0.00	98.85	1.15
9	100.00	0.00	98.60	1.40
8	100.00	0.00	98.54	1.46
7	100.00	0.00	98.29	1.71
6	100.00	0.00	98.32	1.68
5	100.00	0.00	98.30	1.70
4	100.00	0.00	98.45	1.55
3	100.00	0.00	86.13	13.87
2	100.00	0.00	86.13	13.87
Testing phase 12 features	100.00	0.00	99.72	0.28

research, the IDS is applied to HTTP traffic, and the experimental results are shown in Table 8 and Fig. 5.

Table 8 shows the steps of applying the proposed IDS, steps 1–20 are executed in the training phase, and only step 21 is executed in the testing phase. In addition, just one step will be executed in the running phase. In step 0, the HTTP training data set with 38 features are used in both training and testing commands. After each step, the features that cause an error message are excluded (for example, example feature 4, 5, 6, 8, 12 13, 14, 15, 17, 18 and 19 are excluded after step 0). Furthermore, the records that cause FP or FN are excluded (for example, example 344 normal records that cause FP are removed after step 2). After step 20, the FP = 0 and the FN = 0; therefore, the training phase ended and the resultant data set is used as training data set in the testing phase in step 21. The training data set includes only 20 features after training phase. In the testing phase, the testing HTTP data set with the same 20 features is used in testing command to check the performance of

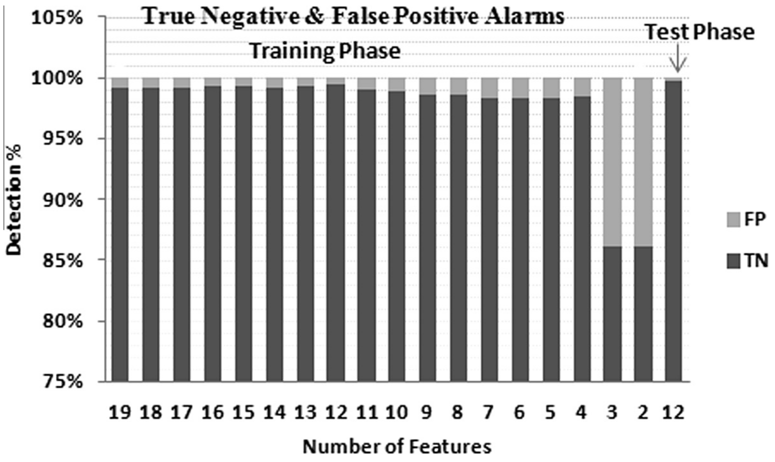

Figure 7 Experimental results HTTP based IDS for Neptune attack.

Table 11 Experimental results HTTP based IDS for Back attack.

# of Features	True positive (%)	False negative (%)	True negative (%)	False positive (%)
21	100.00	0.00	99.84	0.16
20	100.00	0.00	99.84	0.16
19	100.00	0.00	99.84	0.16
18	100.00	0.00	99.84	0.16
17	100.00	0.00	99.84	0.16
16	100.00	0.00	99.84	0.16
15	100.00	0.00	99.84	0.16
14	100.00	0.00	99.84	0.16
13	100.00	0.00	99.84	0.16
12	100.00	0.00	99.84	0.16
11	100.00	0.00	99.84	0.16
10	100.00	0.00	99.84	0.16
9	100.00	0.00	99.84	0.16
8	100.00	0.00	99.84	0.16
7	100.00	0.00	99.84	0.16
6	100.00	0.00	99.84	0.16
5	100.00	0.00	99.84	0.16
4	100.00	0.00	99.87	0.13
3	100.00	0.00	99.90	0.10
2	100.00	0.00	99.88	0.12
1	100.00	0.00	99.88	0.12

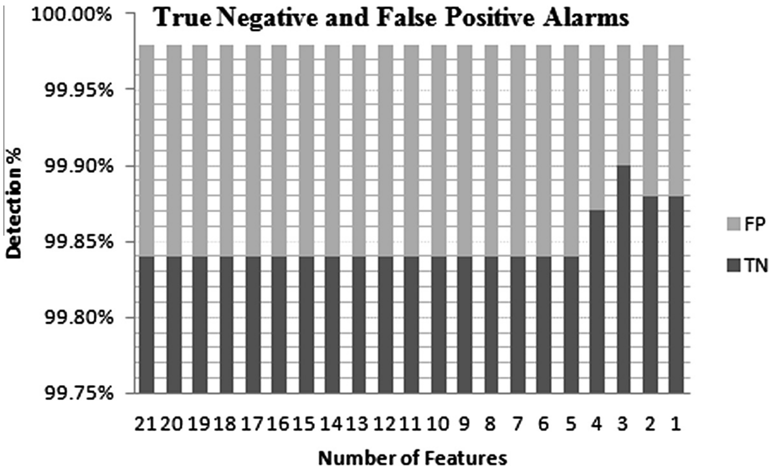

Figure 8 Experimental results HTTP based IDS for Back attack.

Table 12 KDD-test file traffics.

Traffic type	# of Records
Neptune	79
Back	359
Portsweep	2
Saint	1
apache2	737
PH	2
Total attacks	1180
Normal	6673

Table 13 Proposed HTTP based IDS experimental results based on 13 features.

Traffic type	TP (%)	FN (%)	TN (%)	FP (%)
Normal			99.4	0.6
Neptune	100	0		
Back	100	0		
Portsweep	100	0		
Saint	100	0		
apache2	95	5		
Detection rate	99.03			

the IDS. The FN = 2.73% and the FP = 2.54%, these results show improvement of the FP and FN using classified data set over using unclassified data set (FP = 4.61%, FN = 24.06%).

Table 9 and Fig. 6 show the IDS performance results using a full training data set and using the reduced one. The reduced data set did not include the records that cause FP or FN. Clearly after excluding records that cause false alarm the FP is reduced while the FN is increased, in addition reducing the number of features from 27 into 20 features increases FP and reducing FN. The best FN is 0.96% when using the training data set that includes 20 features without excluding any data record. The most important IDS target is to minimize the FN as possible; therefore, it is proposed to use classified training data set without excluding any data record.

4.3. The proposed classified HTTP based IDS

The final HTTP based IDS proposal is to do one more classification level and to select a minimum number of features. As shown in Fig. 3 the HTTP training data set is classified based on an attack type, then the most important features that achieve the optimal IDS performances are selected. Table 10 and Fig. 7 shows some experimental results for HTTP Neptune attack, in training phase the optimal IDS performance could be achieved using 12 features, the FN = 0% and the FP = 0.59%. Using the same 12 features in the testing phase could achieve 0% FN and 0.28% FP. The 12 features that

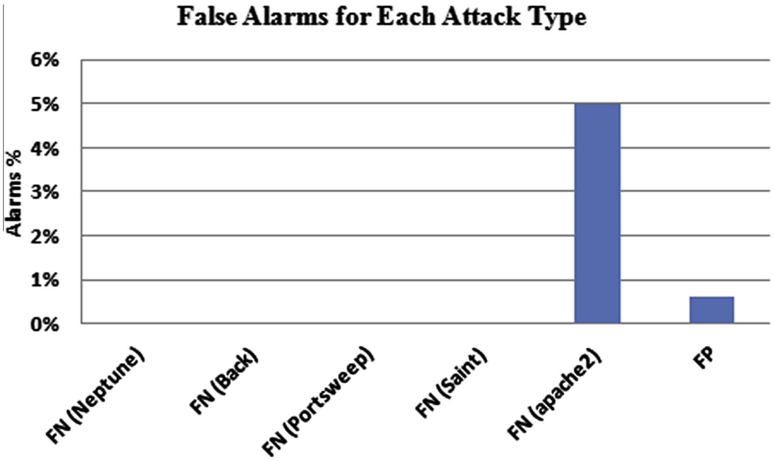

Figure 9 Proposed HTTP based IDS experimental results based on 19 features.

Table 14 Comparison of IDSs detection rates.

IDS index	Intrusion Detection Systems	Detection rate (%)
IDS I	Anomaly based IDS [7]	68.7
IDS II	IDS based on integration of Snort and Bays theorem [6]	96
IDS III	Tree based IDS classification algorithms [5]	97.49
IDS IV	IDS based on data mining techniques [12]	97.5
IDS V	k-Means, K-nearest neighbor and Naive Bayes classifier [3]	98.18
IDS proposed I	The first proposed IDS (Section 4.1)	86
IDS proposed II	The second proposed IDS (Section 4.2)	97.5
IDS proposed III	The proposed classified HTTP based IDS (Section 4.1)	99.03

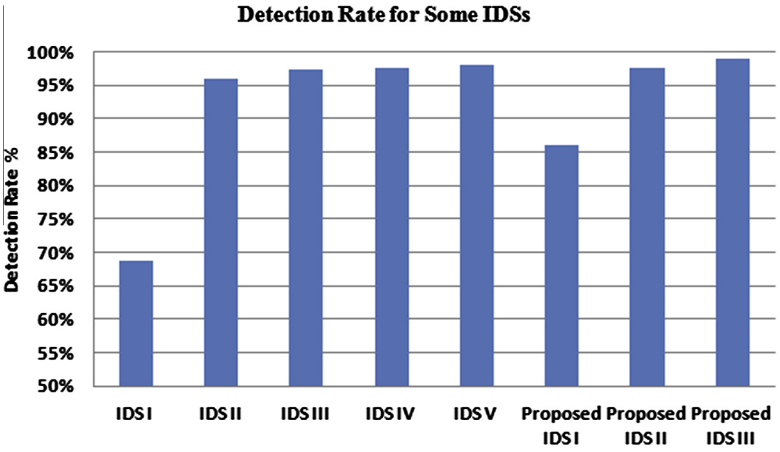

Figure 10 Comparison of IDSs detection rates.

achieve the best IDS performances are: 4 and features from 23 to 33 (see Table 1 in feature's description).

Some HTTP based IDS experimental results for Back attack are illustrated in Table 11 and Fig. 8; the best IDS performance could be achieved using feature 5 (src-bytes), the FN = 0% and the FP = 0.12%.

Finally, the performance of the proposed HTTP based IDS is measured based on the thirteen features: the twelve features that characterize the Neptune attack and the one feature that characterizes the Back attack (Features 4, 5 and features from 23 to 33). The KDD-Test file includes many types of normal and attack traffics; Table 12 illustrates these traffics. The experimental results are shown in Table 13 and Fig. 9, the FP = 0.6%, and FN = 0% for Neptune, Back, Portsweep and Saint attacks, while FN = 5% for apache2 attack, the overall detection rate = 98.96%. Please note that the IDS was only trained on Neptune and Back traffic; however, it not only detects 100% of the Neptune and Back attacks, but it also detects 100% of the Portswep and Saint attacks and 95% of appache2 attacks. The proposed IDS is updated by training on the appache2 attack; the performance was enhanced, the TP = 99.75%, TN = 99%, FP = 1%, FN = 0.25% and the detection rate = 99.03%.

Comparing with similar IDSs; the proposed classified HTTP based IDS could achieve a very good detection rate. Table 14 and Fig. 10 show detection rates for the three proposed IDSs and some of similar IDSs.

5. Conclusions and future works

In this paper, the IDS based on Naive Bayes classifier is analyzed, NSL_KDD data set is used to train and test the IDS, it achieves about 37% FN and 6.6% FP. The first proposal to enhance the IDS performance is preparing the training data set such that it could achieve 100% IDS performance. After applying the IDS for a testing data set, the performance has been improved, the FN is about 24%, and the FP is 4.61%. The target of the second IDS proposal is to improve the performance and to reduce the number of features by selecting only the most important features that characterize each attack type and normal connections, in addition it proposes to classify the data set based on services. In this paper, the IDS aims to detect malicious connections that exploit the HTTP service; therefore, it is categorized as HTTP based IDS. The performance was significantly enhanced, the detection rate = 99.03%, TP = 99.75%, TN = 99%, FP = 1% and FN = 0.25%. While on average, some of the leading IDSs achieve 96% detection rate and 1.5% false-positive rate (see Section 2), therefore; the proposed IDS shows the superiority over similar IDSs.

As a future work, the proposed IDS can be used in the IDS running phase by installing it on a network to protect this network against real time attacks. In addition, the proposed IDS can be applied to other network services such as FTP and IMAP services.

References

[1] Dewan F, Mohammad R, Chowdhury R. Adaptive intrusion detection based on boosting and Naïve Bayesian classifier. Int J Comput Appl 2011;24(3):12–9.

[2] Yang L. The research of Bayesian classifier algorithms in intrusion detection system. In: IEEE international conference on E-Business and E-Government, ICEE 2010. Guangzhou, China; May 2010. p. 2174–8.

[3] Hari O, Aritra K. A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. In: 1st IEEE international conference on recent advances in information technology, RAIT 2012. Dhanbad, India; March 2012. p. 131–6.

[4] Alexander T, Aleksey P, So Grigory. Efficient computer network anomaly detection by change point detection methods. IEEE J Sel Top Signal Process 2013;7(1):4–11.

[5] Sumaiya T, Aswani C. An analysis of supervised tree based classifiers for intrusion detection system. In: IEEE proceedings of the international conference on pattern recognition, informatics and mobile engineering, PRIME 2013. Salem, India; February 2013. p. 294–9.

[6] Chirag M, Dhiren P. Bayesian classifier and snort based network intrusion detection system in cloud computing. In: The third IEEE international conference on computing communication & networking technologies, ICCCNT 2012. Coimbatore, India; July 2012. p. 1–7.

[7] Nabil A, Soroush H, Ljiljana T. Feature selection for classification of BGP anomalies using Bayesian models. In: Proceedings of the international conference on machine learning and cybernetics. ICMLC 2012. Xian, Shaanxi, China; July 2012. p. 140–7.

[8] Quanquan G, Zhenhui L, Jiawei H. Generalized fisher score for feature selection. In: The 27th conf. on uncertainty in artificial intelligence. Barcelona, Spain; July 2011. p. 266–73.

[9] Wang J, Chen X, Gao W. Online selecting discriminative tracking features using particle filter. In Proc computer vision and pattern recognition, San Diego, CA, USA; June 2005, vol. 2, p. 1037–42.

[10] Peng H, Fulmi L, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27(8):1226–38.

[11] Jingnian C, Houkuan H, Shengfeng T, Youli Q. Feature selection for text classification with naive Bayes. Expert Syst Appl 2009;36(3):5432–5.

[12] Chandrasekhar M, Raghuveer K. Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. In: IEEE international conference on computer communication and informatics. Coimbatore, India; January 2013. p. 1–7.

[13] The NSL-KDD data set. <http://nsl.cs.unb.ca/NSL-KDD/> [1.10.13].

[14] KDD Cup 1999 Data. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> [1.10.13].