ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, January 2015

Network Intrusion Detection Using Semi-Supervised Learning Based on Normal Behaviour's Standard Deviation

Tawfiq S. Barhoom [1], **Ramzi A. Matar** [2]

Associate Professor, Department of Information Technology, Islamic University of Gaza, Gaza, Palestine [1]

Department of Information Technology, Islamic University of Gaza, Gaza, Palestine [2]

Abstract: Misuse detection is the traditional technique used in Network Intrusion Detection Systems (NIDSs) which relies on matching the current behavior of network with pre-defined attacks' signatures. This technique is effective to detect the majority of known attacks, but fails to protect from unknown threats, such as zero-day exploits. In addition the increasing diversity and polymorphism of network attacks further obstruct modeling signatures, such that there is a high demand for alternative detection techniques. Many researchers are still trying to solve the problem by using new machine learning techniques such as supervised or unsupervised learning; however producing labeled dataset for supervised learning is difficult, also it is difficult to label the generated clusters to normal or abnormal in unsupervised learning. To overcome these issues we have proposed a novel technique by using semi-supervised learning technique which based on the standard deviation of the normal behavior by which we attempt to detect attacks by calculating their deviations from the normal cluster in observed data.

Keywords: Network Intrusion Detection, Anomaly detection, Semi-supervised learning, Standard Deviation.

I. INTRODUCTION

In the modern life, computer and network play a critical role in people's life. In the world of computer the prevention, detection and respond are the 3 layers of actions to an attack for increasing Confidentiality, Integrity and Availability (CIA) of data and services. The goal of prevention stage is to block any unauthorized access to the network and systems; however it is impossible to block all types of malicious activities. The goal of Intrusion Detection System (IDS) is to detect the attack and send an alarm to the administrator or respond to the attack according to the predefined rules.

Network intrusion attacks are any abnormal behavior try to violate the CIA principles. The network intrusions are divided mainly into four categories:
(1). DOS: Denial of service – where an attacker tries to prevent legitimate users from using a service. e.g. Syn flooding
(2). Probing: Surveillance and other probing, where an attacker tries to gain information about the target host., e.g. port scanning.
(3). U2R: unauthorized access to local super user (root) privileges, where an attacker has local access to the victim machine and tries to gain super user privileges., e.g. buffer overflow attacks.
(4). R2L: unauthorized access from a remote machine, where an attacker does not have an account on the victim machine, hence tries to gain access., e.g. password guessing

There are two techniques of detection in NIDS, signature based and anomaly based. In signature based NIDS, the system looks for the characteristics of known network attacks to detect the existence of such attacks, but it fails to detect novel attacks with different characteristics; this failure is known as zero-day attack. Growing number of zero day attacks and the increasing diversity and polymorphism of network attacks made anomaly based NIDS more efficient. By using this way it is possible to detect novel and unknown network attacks without signatures database of known attacks. Today the challenge is to find a way to have fewer false alarms and more detection rate of complex attacks, especially in imbalance network traffic [1, 2].

Machine learning techniques have been used in NIDS and improve the performance of attack detection[3]. There are three categories of machine learning techniques for NIDS are supervised; semi-supervised and unsupervised learning techniques [3, 4]. Supervised learning technique needs to be trained firstly by pre-classified traffic sample to build the classification model and map the behavior of the network to find the difference between normal and abnormal state. The shortcomings of this technique is that the system is trained on the existing attacks, which may fail to detect a novel attacks, also in most circumstances, labeled data is not readily available since it is time consuming and expensive to manually classify it [5-7]. Many researches try to address these problems by using unsupervised learning techniques such as clustering; by using clustering techniques, they try to measure the deviation of the new instances from the different created clusters. Clustering is the process of assigning a set of objects into group or groups (which called cluster) while the objects in the same cluster are more similar (in some way) compare to other objects [3]. But labeling these clusters is a great problem; which cluster should be labeled as normal and which should be labeled as abnormal.

To overcome these problems we proposed a novel

DOI 10.17148/IJARCCE.2015.4184

approach to detect network intrusions based on the assumption that "*The attack traffic is statistically different from normal traffic*" [8, 9]; this approach is known as semi-supervised detection technique which based mainly on the existence of normal behavior's instances. By using semi-supervised learning technique we can detect any deviation from the normal behavior. This technique requires a set of purely normal data. If the normal instances contain traces of intrusions, the algorithm may not detect future instances of these intrusions because it will assume that they are normal. Purely normal data is also very hard to obtain in practice, since it is very hard to guarantee that there are no intrusions when we are collecting network traffic [6].

To overcome the existence of intrusions in the normal data, we follow the following steps:

(1) Sampling the normal instances to acceptable percentage using stratified sampling. By using this method we eliminate infrequent instances which may be some kind of attacks.

(2) Then we have used the Local Outlier Probability (LoOP) proposed by Kriegel et al [10] to detect the abnormal instances in the normal dataset and eliminate all the instances which have an outlier probability greater than 0.5.

(3) After that we have divided the processed dataset into three clusters based on the transport protocol, TCP, UDP and ICMP, after that we have calculated the standard deviation for each of the three clusters. The standard deviation of the normal cluster is used as the cluster radius or the cluster boundary and any new instance which have a distance from the cluster centroid greater than the standard deviation of this cluster, it is labeled as abnormal. The standard deviation is used to eliminate any abnormal behavior in the normal cluster and gives us the normal behavior boundaries.

The standard deviation (SD) (represented by the Greek letter sigma, σ which is the square root of the variance σ^2) measures the amount of variation or dispersion from the average [11]. A low standard deviation indicates that the data points tend to be very close to the mean (also called expected value); a high standard deviation indicates that the data points are spread out over a large range of values. The variance σ2 is the average of the squared differences from the Mean. There are two formulas to calculate the standard deviation. The "Population Standard Deviation", which is used when we have a complete dataset and the "Sample Standard Deviation" as shown in Formula 1, used when we have a sample dataset. In our proposed method for calculating the standard deviation we used the sample standard deviation because we don't have the a complete normal data, we use the 10% of the normal data.

We evaluated our approach over real network data. Both the training and testing was done using the KDD Cup 1999 data [12], which is a very popular and widely used intrusion attack data set, and It is also widely used and accepted in the academic community.

II. RELATED WORKS

Several recent researches in the few last years were proposed and presented for detecting intrusions in network using both supervised and unsupervised techniques.

A. *Supervised intrusion detection approaches*

Sarnsuwan, Charnsripinyo et al. 2010 [13] provided a new approach to detect internet worm. They considered behaviors of internet worm that is different from the normal pattern of internet activities. Their network features mainly consist of characteristics of IP address, port, protocol and some flags of packet header collected in 1 minute window. These features are used to detect and classify behavior of internet worm by using 3 different data mining algorithms which are Bayesian Network, Decision Tree and Random Forest. They assume that the worm connections were expected to have high number of failure connections. Moreover, the failure connections can be occurred when a source IP sends a request connection packet to an unused IP address or some ports that no longer in service. After that, ICMP packet, SYN/ACK packet and TCP RESET will be returned. So the number of these packets will be high. The approach provided good results with detection rate over 99.6 percent and false alarm rate is close to zero with Random forest algorithm. In addition, the model can classify behaviors of DoS and Port Scan attacks with detection rate higher than 98 percent and false alarm rate equal to zero. Their assumption may failed to detect Camouflaging Worm such as Atak worm [14] and also if a novel worm exists the model may not detect it because it is learned on specific worms, besides that the complexity of labeling the dataset.

Barhoom and Qeshta 2013 [15] proposed a new approach based on data mining techniques for worm's detection; using a combination of classifiers (Naïve Bayes, Decision Tree, and Artificial Neural Network) in order to be adaptive for detecting known/unknown worms, to achieve higher accuracies and detection rate, and lower classification error rate. The results show that the proposed model has achieved higher accuracies and detection rates of classification, where detection known worms are at least 98.30%, with classification error rate 1.70%, while the unknown worm detection rate is about 97.99%, with classification error rate 2.01%. The problem of this model that it was trained on existing worms but it can't detect worms with different behaviors and also the data set doesn't contains any information about the number of network connection failures in a time window which are important for worm detection and classification.

B. *Unsupervised intrusion detection approaches*

Portony et-al [16] presented a method for clustering similar data instances together and uses distance metrics on clusters to determine an anomaly. The author makes two basic assumptions: First, data instances having the same classification should be closed to each other in feature space under some reasonable metric, while instances with different classifications should be far apart. Second, the number of instances in the training set that represent normal traffic is overwhelmingly larger than the

DOI 10.17148/IJARCCE.2015.4184

number of intrusion instances. Clusters were labeled based on cluster size; the biggest cluster (>98%) will be labeled as normal and others as abnormal. The solution is able to detect new types of intrusion while maintaining a low false positive rate. Their method is effective when almost network traffic is normal class and homogenous, but the problem of this solution is that they depend on one technique which is 'size', which may be not accurate in DoS attacks, in which almost data is abnormal, the big cluster (actually abnormal) will be considered as normal. Also if any assumption doesn't achieve its criteria, the system accuracy will decrease and give high false alert.
Bhuyan et-al [16] used a new solution which detects network anomalies using an unsupervised approach with minimum false alarms. First, they introduce a tree based subspace clustering technique for generating clusters in high dimensional large datasets. Their approach exploits a specific technique for finding a highly relevant feature set. Second, they analyze the stability of the cluster results obtained. Third, they propose a cluster labeling technique to label the stable clusters using a multi-objective approach using cluster size, compactness and dominating feature subset. The solution used multi approaches for labeling the clusters; it will decrease false alarm, while increase the percent of detection rate. The problem in this solution is that stability of cluster is not exclusive in normal clusters, but also in abnormal clusters such as DoS. In addition, they didn't determine the techniques that have been used for choosing relative features.

Leung et-al [6] proposed a density based and grid based clustering algorithm, that uses adaptive grid algorithm and FP-tree growth method for frequent item set mining. They aim to discover clusters from large volume of high dimensional input data. Grid-based methods divide the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure. Once they obtain the set of clusters, they expect that they cover most but not all of the data set. Therefore any point that falls inside the clusters will be labeled as normal. The small percentages of points that do not belong to any clusters are labeled as abnormal. Their solution has the advantage that it can produce clusters of any arbitrary shapes and cover over 95% of the data set with appropriate values of parameters. They have evaluated the accuracy of the new approach and showed that it achieves a reasonable detection rate while maintaining a low positive rate. The problem is that they consider the large cluster as normal, but if there is any difference or changes in this assumption, the accuracy will be decreased and system will give high false alert. In addition, they assume a small percentage of points that do not belong with any clusters are labeled as abnormal, but in the real network this is not always true, all points must belong to the clusters. Another problem is the consuming time for extracting frequent item sets from high dimensional feature space.

Jiang, Song et al. [17] considered the outlier factor of clusters for measuring the deviation degree of a cluster in order to detect intrusions attacks. The authors proposed a novel method to compute the cluster radius threshold, which is the threshold of the maximum distance between all the points and the cluster centroid. The data classification is performed by an improved nearest neighbor (INN) method to label clusters. The outlier factor of cluster that they defined is used to measure the degree of a cluster deviating from the whole where anomalous classes can be distinguished from normal ones. They obtained an improved nearest neighbor (INN) method for classifying data and a novel strategy for detecting intrusion. INN considers not only the candidate classified object and its nearest neighbor in model, but also the distance between them. The proposed intrusion detection approach can theoretically detect new types of attacks. The proposed strategy achieves both higher detection rate and lower false alarm rate than previous methods. In particular, it is capable of detecting unknown intrusions. The method is composed of three parts: First, creating clusters from unlabeled training datasets; second, labeling clusters as 'normal' or 'anomalous' by their outlier factors; and third using the labeled clusters to classify network data. Based on the strategy of labeling, it failed to label some attacks such as R2L attacks as abnormal.

III. APPROACH & METHODOLOGY

We proposed this approach in detecting network intrusions based on the assumption: that in order to differentiate between abnormal activities and normal activities we need to learn first the normal activities to be able to identify any abnormal activities. This assumption is essentially in any learning methodology. The challenges of applying this assumption in networks is difficult because we can't guarantee that the existing normal activity is absolutely free from any type of attacks specially R2L attacks which have a behaviour near the behaviour of normal activities.

To overcome this issue we have proposed a new approach in which we will be highly guaranteed that the normal activity is free from any type of attacks. This novel approach to detect network intrusions is based on the assumption that "The attack traffic is statistically different from normal traffic" [8, 9]. This approach is a semi-supervised detection technique which based mainly on the existence of normal behaviour data. To overcome the existence of intrusions in the normal data we have followed these steps by which we eliminate the infrequent instances and the instances that is 50% to be outliers:

To overcome the existence of intrusions in the normal data, we follow the following steps:

(1) Sampling the normal instances to acceptable percentage using stratified sampling. By using this method we eliminate infrequent instances which may be some kind of attacks.

(2) Then we have used the Local Outlier Probability (LoOP) proposed by Kriegel et al [10] to detect the abnormal instances in the normal dataset and eliminate all the instances which have an outlier probability greater than 0.5.

DOI 10.17148/IJARCCE.2015.4184

(3) After that we have divided the processed dataset into three clusters based on the transport protocol, TCP, UDP and ICMP, after that we have calculated the standard deviation for each of the three clusters. The standard deviation of the normal cluster is used as the cluster radius or the cluster boundary and any new instance which have a distance from the cluster centroid greater than the standard deviation of this cluster, it is labeled as abnormal. The standard deviation is used to eliminate any abnormal behavior in the normal cluster and gives us the normal behavior boundaries.

As illustrated in Fig. 1, there are 3 clusters, the cluster with circle instances, which is the biggest scattered one, is the normal cluster and the others, with square and triangle instances, are the abnormal clusters. The normal boundary from the cluster center is the first circle which is the standard deviation of it and any expanding of this boundary will decrease both the false alarm and the intrusion detection rate. As shown in Fig. 1, we need to adjust the cluster boundary in order to achieve high intrusion detection rate and low false alarm rate, and to do so, we have added a new parameter named as scalar parameter, which is used as a scalar added to the standard deviation's of the normal cluster in order to expand the cluster boundaries.

The three created clusters in the used dataset are generated based on the transport protocol type, TCP, UDP and ICMP. The purpose of creating these clusters is that each transport protocol has its own behavior's characteristics, which means that each cluster has its own feature space that differs than the other clusters, in addition to the common relative features. For example TCP protocol has a session period feature which doesn't exists in the other two protocols UDP and ICMP. After generating the three clusters, the standard deviation of each cluster is calculated based on its centroid.

By this way we gain many benefits;

(1) Saving time of distance measurements, we don't have to measure the distance between a new UDP instance and the TCP cluster, where the time needed to measure the distance from TCP cluster differ than the time needed to measure the distance from UDP or ICMP cluster due to different feature sets.

(2) Accurate distance measure, we don’t have to calculate the distance of irrelative attributes like period time in TCP which is not exists in both UDP and ICMP instances.

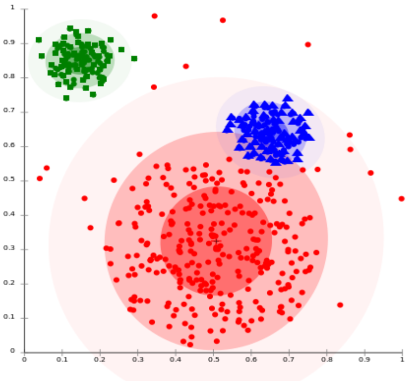

Fig. 1 The circle instances cluster is the normal behavior with standard deviation from center to the first inner circle.

IV. DATASET

We evaluate our approach using network real data known as KDD Cup 1999 dataset [12] which was prepared and managed by MIT Lincoln Labs. This dataset is used as a benchmarking for intrusion detection systems, and it is widely used and accepted in the academic community.

The training data is made up of 22 different attacks out of the 39 present in the test data. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test datasets not available in the training data sets.

The training dataset consisted of 494,021 records among which 97,277 (19.69%) were normal, 391,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R connections. In each connection there are 41 attributes describing different features of the connection and a label assigned to each either as an attack type or as normal. We used the normal data, extracted from 10% training dataset, to build our model and evaluating the model using the 10% testing dataset which means that all attacks are new to our model because our model didn't trained on them.

There are multiple attack types for each category as shown in table 1 [12], each attack has its own characteristics and behavior on network. Our system detects most of these attacks without training it on them.

TABLE 1
ATTACKS CATEGORIES

Category	Type
DoS	smurf, neptune, back, teardrop, pod, land
Probe	satan, ipsweep, portsweep, nmap
R2L	warezclient, guess_passwd, warezmaster, ftp_write, multihop, phf, spy, imap
U2R	buffer_overflow,rootkit, loadmodule, perl

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic.

V. EXPERIMENTAL SETUP

(1) Extracting a sample of normal instances by stratified sampling of the 10% normal dataset to eliminate the infrequent instances which may be outliers or abnormal activities. The number of instances of the normal dataset after sampling was 12000 records.

(2) After that we have converted the polynominal attributes into numerical and normalized the feature src_byte and dst_byte in the range from 0 to 9 and normalized the attributes dst_host_count, dst_srv_count, srv_count, count, and duration in the range from 0 to 1.

(3) Then we have computed the information gain to determine the relative features that is needed to build our mode.

(4) After that we have used the LoOP [10] to detect the abnormal instances in the normal dataset and eliminate all the instances which have an outlier probability greater than 0.5, in KDD[12] dataset we have sat k=60 with

DOI 10.17148/IJARCCE.2015.4184

normalization factor=3 and then divided the remained dataset into three clusters based on the transport protocol.

(5) Finally we have computed the standard deviation for each cluster based on its centroid.

The standard deviation of each cluster is derived by computing the distance of each instance in the normal dataset, that has the same cluster's transport protocol type, from the cluster centroid using the Euclidean distance then applying the Sample Standard Deviation to get the standard deviation of the cluster as shown in the general Formula 1:

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \qquad \text{Formula (1)}$$

The Euclidean distance can be calculated using the following formula as shown in Formula 2:

$$\mathtt{Xdist} = \sqrt{\sum_{i=1}^{F}(x_i - \mathtt{ci})^2} \qquad \text{Formula (2)}$$

Where Xdist is the distance of instance X from the cluster centroid C, Xi is the feature i of the instance, Ci is the feature i in the cluster centroid and F is the total number of features of the instance based on its protocol type.

After calculating the distances of all the instances in from the desired cluster based on the transport protocol type we use Formula 3 to get the sample standard deviation of the cluster, which is used as the boundary or radius of the cluster.

$$\mathtt{s} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\mathtt{Xdist}_i)^2} \qquad \text{Formula (3)}$$

Where N is the number of all instances depending on their transport protocol type.

We have used RapidMiner Studio 6 to perform stratified sampling and to compute the Local Outlier Probability, and Oracle database is used to built the centroid tables of the three clusters, TCP, UDP, and ICMP, and to perform the detection process. The TCP cluster contains 7741 instances, the UDP cluster contains 1742 instances and the ICMP cluster contains 135 instances.

We have computed the detection rate and false alarm rate for each cluster more than one time, at each time we increment the scalar parameter to expand the cluster boundaries in order to get lower false alarm, the scalar parameter starts from 0. The testing data set that we used contains 311029 in which 119357 instances that used TCP protocol, 26702 instances that used UDP protocol and 164969 instances that used ICMP protocol.

VI. Results

After performing the experiments, we have created six tables, the first three tables, Table 2,3 and 4, are one table for each cluster, listed the detection rates of each attack grouped by their attack type, the last three tables, Table 5,6 and 7, list the detection rate for each attack type. All of the six tables have a header rows, the first header contains the False Alarm rate, which is based on the second row, the scalar parameter. The number of normal instances in all of the six tables is written between two brackets beside the false alarm label. Also the number of attack instances and the number of attack type instances are the same.

Table 2 shows the results of attacks detection which uses TCP in the 10% testing dataset, as we see in this table; we notice that when the scalar parameter is 0 the detection rate of all attacks is approximately 100%, except for some attacks like mailbomb. guess_passwd., zero escalation in standard deviation value using the scalar parameter means that there's no expanding of the normal dataset boundaries which have been evaluated using its standard deviation. On the other side we notice that the false alarm is 4.57 which is large a bit. So we need to lower the false alarm using the scalar parameter. Table 1 listed the scalar parameter value and it's corresponding false alarm and detection rate for each attack.

At each increment value of the standard deviation using the scalar parameter there's a decrease in the false alarm rate as well as in the detection rate, but as we see in table 2 the decrease of the detection rate is slower in DoS and Probe attacks than the R2L and U2R attacks. A visual graph for this decreases is illustrated in Fig.s 2,3 and 4 in the discussion section. The attacks, mailbomb, gues_passwd, and processtable whill be discussed in the discussion section.

TABLE 2
RESULTS OF ATTACKS DETECTION THAT USES TCP PROTOCOL

False Alarm (44118)	**4.57**	**2.53**	**1.16**	**0.89**	**0.75**
Scalar Paramter	**0**	**0.35**	**0.5**	**0.55**	**0.65**
Attack	**DoS**				
apache2.(794)	99.6	99.4	97.5	95.1	86.9
back.(1093)	99.5	99.5	99	98.2	97.7
land.(9)	100	100	100	100	100
mailbomb.(5000)	0.28	0	0	0	0
neptune (58001)	100	100	100	100	100
processtable. (759)	100	46.9	40.7	38.6	37.9
Attack	**Probe**				
mscan. (1053)	100	99.1	96.6	94.5	92.2
nmap. (84)	100	100	100	100	100
portsweep. (354)	100	100	100	100	100
saint. (607)	99.8	99.8	99.7	99.7	99.3
satan. (1219)	100	99.9	99.9	99.9	99.8
Attack	**R2L**				
ftp_write. (3)	100	100	33.3	33.3	33.3
guess_passwd.(4367)	17.1	16.4	6	5.38	4.4
imap. (1)	100	100	100	100	100
multihop. (9)	100	100	100	100	88.9

DOI 10.17148/IJARCCE.2015.4184

named. (17)	100	100	58.8	52.9	35.3
phf. (2)	100	100	100	100	100
sendmail. (17)	100	100	100	94.1	47.1
warezmaster. (1602)	99.9	99.3	91.5	79.7	36.4
worm. (2)	0	0	0	0	0
xlock. (9)	100	88.9	88.9	66.7	22.2
xsnoop. (4)	100	100	100	100	75
Attack	**U2R**				
buffer_overflow.(22)	100	95.5	95.5	95.5	77.3
httptunnel. (158)	100	98.7	93	88.6	86.1
loadmodule. (2)	100	100	100	100	100
perl. (2)	100	100	100	100	100
ps. (16)	87.5	87.5	68.8	62.5	37.5
rootkit. (13)	100	84.6	38.5	38.5	38.5
sqlattack. (2)	100	100	100	100	100
xterm. (13)	100	92.3	76.9	76.9	69.2

Table 3, lists the false alarm and the detection rates of attacks that use the UDP transport protocol. As we see when the scalar parameter is zero the detection rate is approximately 100% and the false alarm is high in the other side. Note that there's attacks that use more than one protocol, these attacks are of type Probe at which an attacker tries to gain information about the target host. e.g. port scanning, which gives information about the running services.

The number of UDP normal instances are 16096 which is large enough to measure the standard deviation of its cluster, as shown in table 3, the udpstorm which belongs to DoS attacks has only 2 instances which are not enough to measure it's detection rate despite its detection rate was 100% at scalar parameter 0, the same thing happened with multihop attack which has only 8 instances, but its detection rate was 100% at 0 and 0.1 scalar parameter values.

The most extreme attacks are R2L attacks, snmpgetattack and snmpguess, which have large number of instances, 7741 and 2403 respectively although our model failed to detect them even when the scalar parameter was 0, this means that their behavior looks like the normal behavior, these attacks exploit the vulnerability of SNMP.

TABLE 3
RESULTS OF ATTACKS DETECTION THAT USE UDP PROTOCOL

False Alarn(16096)	**4.11**	**1.45**	**1.35**	**1.19**
Scalar paramter	**0**	**0.2**	**0.25**	**0.3**
Attack	**DOS**			
teardrop. (12)	100	100	100	100
udpstorm. (2)	100	0	0	0
Attack	**Probe**			
saint. (27)	96.3	85.2	77.8	55.6
satan. (413)	100	99.5	99.3	99.3
Attack	**R2L**			
multihop. (8)	100	0	0	0
snmpgetattack. (7741)	0.16	0	0	0
snmpguess. (2403)	0	0	0	0

Table 4, lists the attacks that use ICMP protocol, we have notice from the results that the number of normal instances are not enough to build the model which is 378 instances.

TABLE 4
RESULTS OF ATTACKS DETECTION THAT USE ICMP PROTOCOL

False Alarn (378)	**61.9**	**18**	**15.1**	**3.97**	**0.53**
Scalar paramter	**0**	**0.25**	**0.3**	**0.35**	**0.6**
Attack	**DOS**				
pod. (87)	100	98.9	98.9	93.1	85.1
smurf. (164091)	100	100	90.9	80.1	80
Attack	**Probe**				
ipsweep. (306)	100	98	98	98	25.2
saint. (102)	100	99	99	99	9.8
satan. (1)	100	100	100	100	0

As we see in the table the false alarm is large, 61.9% at the 0 value of the scalar parameter which means that the normal instances is scattered and have varies distances from the cluster centroid, also we note that the detection rate is more than 99% at scalar parameter values less than 0.3. We will discuss this issue in the discussion section.

The following three tables list the detection rate grouped by the attacks types.
Starting with the first table, Table 5, which lists the detection rate of the attacks types that use the TCP protocol, we excluded mailbomb attack from dos attacks and guess_passwd attack from r2l attacks because both attacks have a large number of instances and a very low detection rate reaches less than 5% when scaling the standard deviation to get false alarm, we excluded them because they affect the overall detection rate because of their large number instances.
As we see, the appropriate scale parameter value is 0.5 at which we gain low false alarm,1.16, and high detection rate, specially for r2l and u2r attacks which their behavior is near the normal behavior.

TABLE 5
TCP DETECTION RESULTS GROUPED BY ATTACK CATEGORY

False Alarn (44118)	**4.57**	**3.16**	**1.16**	**0.89**	**0.71**
Scalar paramter	**0**	**0.3**	**0.5**	**0.55**	**0.75**
DoS (60661)	100	99.4	**99.2**	99.1	98.3
Probe (3317)	100	99.9	**98.8**	98.2	96.4
R2L (1666)	99.8	99.5	**91.1**	79.5	36.4
U2R (228)	99.1	97.4	**87.7**	84.2	78.5

Table 6, lists the detection rate of the attacks types that use the UDP protocol. Based on the results shown in the table, note that we exclude the udpstorm attack from the

DOI 10.17148/IJARCCE.2015.4184

DoS attacks detection rate because it has just two instances which are not enough to evaluate the model accuracy. The Probe attacks has high detection rate but with R2L attacks our model failed to detect them, these attacks are listed in Table 3 which need more investigation about their lower detection rate.

TABLE 6
UDP DETECTION RESULTS GROUPED BY ATTACK CATEGORY

False Alarn(16096)	4.11	2.23	1.45	1.35	0.77
Scalar paramter	**0**	**0.1**	**0.2**	**0.25**	**0.35**
DoS (14)	100	100	100	100	100
Probe (440)	99.8	99.5	98.6	98	95.7
R2L(10152)	0.2	0.2	0	0	0

Table 7, lists the detection accuracy of the attacks types that use the ICMP protocol. The most major problem we noticed from the results is the lower number of instances of the normal instances in the training dataset which was just 135 records, which is not big enough to calculate the standard deviation of its cluster, as shown in the table below, the detection accuracy was high with scalar value less than 0.3 but the false alarm was 37.8% which indicate that the normal instances are scattered and have a varies distances from the ICMP cluster centroid.

TABLE 7
ICMP DETECTION RESULTS GROUPED BY ATTACK CATEGORY

False Alarn (378)	61.9	15.1	3.97	3.44	0.53
Scalar paramter	**0**	**0.3**	**0.35**	**0.45**	**0.6**
dos (164178)	100	90.9	80.1	80.1	80
probe (409)	100	98.3	98.3	47.9	21.3

VII. DISCUSSION

Based on the results of our model we have noticed some of the most interesting things that need to be discussed starting with the relation from detection rate and false alarm. In general as shown in Fig. 2 whenever the false alarm decrease the detection rate also decrease this is due to the existence of extreme normal activities which near the activities of attacks.

Fig. 2 shows that for every increase of the scalar parameter which in order increases the cluster boundaries to surround all the extreme normal activities there's a decrease in the detection rate as shown in Fig. 1. So we need to make a tradeoffs between the false alarm rate and the detection rate.

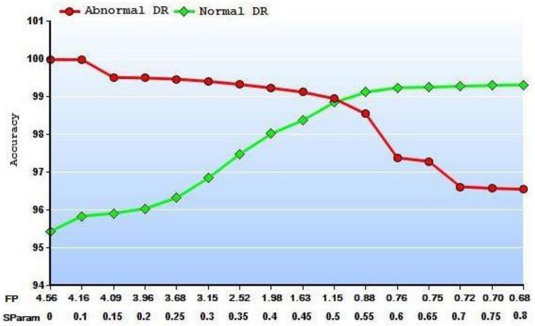

Fig. 2 The relationship between detection rate and false alarm depending on the scalar parameter

For attacks such as R2L which behaves like normal activities in its major characteristics, the detection rate decreased rapidly in any increase of the normal boundaries using the scalar parameter as shown in Fig. 3 the warezmaster attack have been detected with accuracy reached 100% with false alarm greater than 2% and started to decrease rapidly when false alarm less than 2%, as shown in Fig. 3, 88% with false alarm 1.15% and decreased to 39% with false alarm 0.76%.

The other type of attacks such as U2R in Fig. 4, e.g httptunnel attack, there's also a rapid decrease which has slower decrease than R2L attacks but faster decrease than DoS and Probe attacks, for instance looks at Table 5.

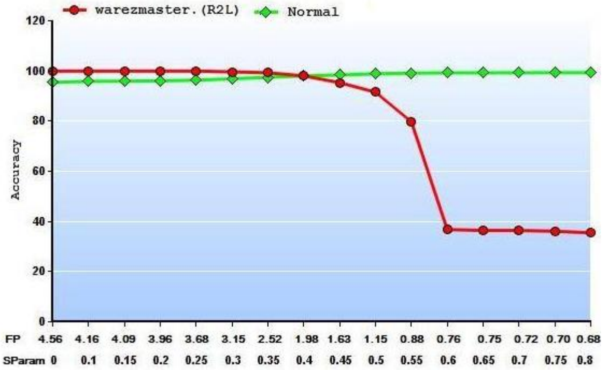

Fig. 3 the relationship between detection rate and false alarm in detecting R2L attacks (e.g. warezmaster attack)

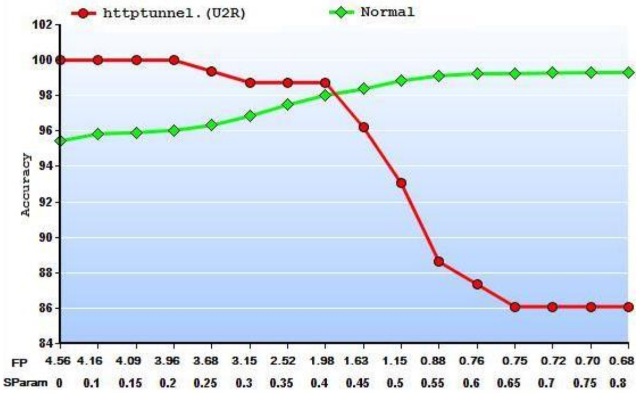

Fig. 4 the relationship between detection rate and false alarm in detecting U2R attacks (e.g. httptunnel attack)

PROBE and DOS attacks of the system are superior to that of other attacks, especially detection of R2L attacks. We analyzed the results in detail and found the reason for the low detection rate for R2L attacks. Both PROBE and DOS attacks often have the distinct traffic characteristic while U2R and R2L are more similar to normal examples. Especially, two R2L attack types (snmpgetattack , snmpguess and guess_passwd) are hardly detected, which account up rough 63% of all R2L attacks. In fact, they are almost identical with normal examples and hardly detected only by the connection information.

VIII. CONCLUSION AND FEATURE WORK

We have proposed a novel semi-supervised intrusion detection approach which gain benefit of supervised learning and unsupervised learning such as classification and clustering respectively. We used standard deviation with a scalar parameter to determine the cluster boundaries and any instance has a distance greater than the standard deviation labeled as abnormal. The results show that our approach has the ability to detect new attacks with high detection rate and low false alarm rate.

We have observed some issues which we aim to address them in our feature work, these issues are:

(1) Automatic adjusting the scalar parameter taking in account to balance between the detection rate and false alarm rate.

(2) Try to divide the normal instances in the training dataset into 3 dataset depending on the transport protocol in order to balance the sampling process and outlier detection through making a stratified sample for each of them and apply the LoOP for each of them to overcome the case of ICMP low instances which its results appeared in Table 7 at which its cluster size was just 135 instances.

REFERENCES

[1] Sperotto, A., et al., *An overview of IP flow-based intrusion detection.* Communications Surveys & Tutorials, IEEE, 2010. **12**(3): p. 343-356.

[2] Engen, V., Machine learning for network based intrusion detection: an investigation into discrepancies in findings with the KDD cup'99 data set and multi-objective evolution of neural network classifier ensembles from imbalanced data, 2010, Bournemouth University.

[3] Nguyen, T.T. and G. Armitage, *A survey of techniques for internet traffic classification using machine learning.* Communications Surveys & Tutorials, IEEE, 2008. **10**(4): p. 56-76.

[4] Bhuyan, M.H., D. Bhattacharyya, and J.K. Kalita. An effective unsupervised network anomaly detection method. in Proceedings of the International Conference on Advances in Computing, Communications and Informatics. 2012. ACM.

[5] Hameed, S.M. and S.S. Sulaiman, *Intrusion Detection Using a Mixed Features Fuzzy Clustering Algorithm.* Iraq Journal of Science (IJS), 2012. **53**(2).

[6] Leung, K. and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. in Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38. 2005. Australian Computer Society, Inc.

[7] Amoli, P.V. and T. Hamalainen. Real time multi stage unsupervised intelligent engine for NIDS to enhance detection rate of unknown attacks. in Information Science and Technology (ICIST), 2013 International Conference on. 2013. IEEE.

[8] Javitz, H.S.V., A., *The NIDES statistical component: Description and justication.* Technical report, SRI International., 1993.

[9] Denning, D., *An intrusion detection model.* In IEEE Transactions on Software Engineering 13., 1987.

[10] Kriegel, H.-P., P. Kröger, and A. Zimek. Outlier detection techniques. in Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2009.

[11] Bland, J.M.A., D.G., Statistics notes: measurement error. 1996.

[12] 12. KDD, The third international knowledge discovery and data mining tools competition dataset (KDD99 Cup). *http://kdd.ics.uci.edu/databases/kddcup99/*. 999 access 24/12/2014.

[13] Sarnsuwan, N., C. Charnsripinyo, and N. Wattanapongsakorn. A new approach for internet worm detection and classification. in Networked Computing (INC), 2010 6th International Conference on. 2010. IEEE.

[14] Yu, W., et al., *Modeling and detection of camouflaging worm.* Dependable and Secure Computing, IEEE Transactions on, 2011. **8**(3): p. 377-390.

[15] Barhoom, T.S. and H.A. Qeshta. Adaptive Worm Detection Model Based on Multi Classifiers. in Information and Communication Technology (PICICT), 2013 Palestinian International Conference on. 2013. IEEE.

[16] Portnoy, L., Intrusion detection with unlabeled data using clustering. 2000.

[17] 17. Jiang, S., et al., *A clustering-based method for unsupervised intrusion detections.* Pattern Recognition Letters, 2006. **27**(7): p. 802-810.

DOI 10.17148/IJARCCE.2015.4184