Network Intrusion Detection Using One-Class Classification Based on Standard Deviation of Service's Normal Behavior

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

[12] *IT. Department, Islamic University of Gaza, Palestine*

Abstract- *A lot of efforts have been given toward designing a perfect NIDS that has a high detection rate and low false alarm rate. Some have used misuse detection technique which fails to detect zero-day attacks, while the problem of using supervised learning is the cost of producing labeled dataset which is essential for training the model and also the model is trained on known attacks which may fail to detect new variant attacks. On the other hand, unsupervised learning has the problem of labeling the generated clusters. Once-Class Classification learning technique (OCC) suffers from the high dimensional network feature spaces, Also, problems may arise when large differences in density exist. To overcome these problems, we proposed OCC-NIDS model based on the standard deviation of service's normal behaviour. Through this model we dealt with each network service as single class instead of dealing with all network services as a single class. By this way we use just the relevant features of each service, hence reducing the high dimensional network feature spaces and also ensure that each class has - a proximately - uniform distribution. The proposed model proved that it is able to detect abnormal network traffic with high detection rate and low false positive rate. It achieved 99.72% detection rate and 99.65% accuracy rate with a false alarm rate reached 0.7% and false positive rate 0.005% on KDD Cup'99 dataset.*

Keywords: *Network Intrusion Detection, Service's Normal Behaviour, One-Class Classification, Standard Deviation*

I. INTRODUCTION

In the modern life, information technology and communications infrastructure play a critical role in people's life. The Internet connects thousands of sub-networks and thereby links over 1billion computers worldwide [1]. The variety of attacks affected computers linked to the Internet, ranging from zero-day exploits crafted for stealthy compromises to computer worms capable of mass-infections. These attacks put both personal as well as business computer systems at risk to be remotely compromised and misused for illegal purpose, (e.g., gathering of confidential data, affecting services availability or violating data integrity which are the three main components of computer security known as confidential, Integrity, Availability (CIA) Triad).

There are two main problems that cause the increase of networks attacks: First, there is a deficit of security awareness in software development [2], (e.g. existing of bugs which make it a vulnerable for attacks exploitations like stake-overflow). A second reason is due to the increasing automation and sophistication of network attacks [3]. A widespread availability of generic attack tools that have an amazing range of functionality, including network surveillance, polymorphic shellcodes and distributed propagation. As an example, the computer worm "Slammer" possess the ability to infect thousands of hosts in a couple of minutes [4]. Such capabilities make malicious software and network attacks attractive for illegal business, as they allow for abuse of millions of computer systems. Due to the explosive growth of the network attacks, intrusion detection systems have become an essential network component which plays a vital role for computer networks and security.

An intrusion is defined by Heady et al. [5] as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. Intrusion detection systems types divided mainly based on their scope into two main types, network based (NIDS) and host based (HIDS) intrusion detection systems [6]. Network Intrusion Detection Systems (NIDS) are placed at a strategic network point or points within the network to monitor and analyze the traffic come from or to all devices on the network in order to detect any illegal/abnormal activity. Whereas Host Intrusion Detection Systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and notify the user or administrator if suspicious activity is detected. Our approach is a NIDS.

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 major techniques of detection in NIDS, signature based and anomaly based. In signature based NIDS, the system looks for the characteristics of

known network attacks, stored in its own database, 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 higher detection rate of complex attacks, especially in imbalance network traffic [7, 8]. Our proposed approach is an anomaly detection technique which based on measuring the deviation of any network instance from the normal behavior of the used service using the standard deviation.

Machine learning techniques have been used in anomaly based NIDS and improve the performance of attack detection [9]. There are mainly three categories of machine learning techniques for NIDS which are supervised; semi-supervised and unsupervised learning techniques [9-11] in addition to OCC [12]. 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 variant attacks [13], also in most circumstances, labeled data is not readily available since it is time consuming and expensive to manually classify it [14-16]. In many practical applications there are a massive data which are often unlabeled like mail spam. The limited labeled data are not enough to train a supervised classifier with fine generalization performance.[17].

Many researchers have tried to address these problems by using unsupervised learning techniques such as clustering [15, 16, 18]; by using clustering techniques, they try to measure the deviation of the new instances from the different created clusters. But labeling these clusters is a great problem; which cluster should be labeled as normal and which should be labeled as abnormal [19]. Laskov, Düssel et al. [13] carried out an experimental framework for comparative study of various supervised and unsupervised approaches for intrusion detection. Their results indicate that the problem of unlabeled data being drawn from a different distribution remain unsolved within the purely supervised or unsupervised techniques and they put their marks on semi-supervised learning approaches that it may provide the superior intrusion detection ability.

To overcome the shortcomings of supervised and unsupervised learning techniques, semi-supervised learning is being used [20]. This technique exploits unlabeled data in addition to labeled ones. Many researchers have used this technique in intrusion detection [11, 17, 21]. Although this learning technique solved the problem of labeling instances and gain the ability of prediction based on relatively a few labeled examples, it suffers from the limitation that it cannot outperform supervised classification unless the analyst is absolutely certain that there is some nontrivial relationship between labeled and the unlabeled distribution [22]. It is also well known that the utilization of unlabeled dataset U is not always helpful for semi-supervised learning algorithms. In particular, it is not guaranteed that adding U to the training data, T, which has a labeled instances L i.e., $T = L \cup U$, leads to a situation in which we can improve the classification performance [2, 22]. When Semi-Supervised learning assumptions are made, but do not hold, it can degrade the performance and can be worse than supervised learning [22].

Because of the previous mentioned detection techniques limitations and shortcomings, and because of the increased diversity of attacks that we can't predict its future behavior, an alternative detection technique that can overcome these obstacles is needed. So, we need a machine learning technique that learns just the normal behavior and detect any deviation from it. This technique is known as One-Class Classification (OCC). Because of the increasing diversity and polymorphism of network attacks which means that very few of these attacks are known, or they do not form a statistically-representative sample of the negative concept. So there's an urgent need to learn how the positive class behave to detect any deviation from it which may be a negative class. Many algorithms for intrusion detection based on OCC have been propose [11, 23-26], many of them have used One Class Support Vector Machines (OCSVMs) which is based on Gaussian Kernel function. Others have used other techniques such as v-SVC [26] and standard deviation [11]. Most of the proposed NIDSs that have applied OCC deal with the whole network instances as a single class, so their proposed NIDs suffer from the high dimensional network feature spaces, and also from the existence of large differences in density which affect the detection accuracy. As far as we know almost all of them have not considered to detect attacks based on the standard deviation of normal behavior of the used service such as HTTP service.

To overcome these challenging issues in OCC, we propose OCC-NIDS model based on the standard deviation of network service's normal behavior. Through this model we deal with each network service as single class instead of dealing with all network services as a single class. By this way we just use the relevant features of each service, hence reducing the high dimensional network feature spaces and also ensure that each class has - a proximately - uniform

distribution.

The authors in [11], used the standard deviation in detecting the deviation of new network instances from its same transport protocol's normal behavior. But they face a problem of large differences in density within a single transport protocol class, which limited them from detecting some attacks. Also the feature space of one class was high because of the existence of all services features, and this also affected the distance measurements because of calculating the distance between a new service instance with irrelevant features that belong to other service. (e.g. feature to tell if SMTP instance initiate communication with HELO message is irrelevant to other services).

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.

Barhoom and Matar [11] proposed a novel OCC learning technique based on the standard deviation of transport protocol's normal behavior. The transport protocol are TCP, UDP and ICMP. By this technique they measured the deviation of any new instance from the same transport protocol class. The standard deviation of each transport protocol class is being the class radius, if the distance between the new instances and the relative transport protocol's class greater than the class standard deviation then the instance is labeled as abnormal else it is labeled as normal. The experimental results on KDD Cup'99 dataset [27] show high detection rate 87.7%-99.2% with low false alarm 1.16%. This work suffers from the high dimensional feature spaces in each transport protocol's class and also each transport protocol's class has varying density. These problems arise due to the existence of several network services in each class (e.g. HTTP, SMTP in TCP class) which affects the overall detection rate and false alarm rate. Our new primary model is based on the service's normal behavior instead of protocol's normal behavior, hence each service has its own relevant features set.

Araki, Yamaguchi et al. [24] proposed a multistage intrusion detection model based on OCSVM focusing on communication interval. The multistage OCSVM uses three sets of traffic, two sets retrieved from a traffic archive and one extracted from real network. At the first stage, OCSVM learns older archive set and then analyzes newer archive set and one from real network. At the second stage, OCSVM learns outlier traffic from the newer archive set and analyzes that from the real network. As a result, extracted traffic from outlier of the real network which does not exist in the newer set can be extracted. They evaluated their method using Kyoto2006+ [28] dataset and 6 new features. The results show that their method detects attacks with 94% detection rate and 6% false positive rate. The proposed algorithm suffers from high dimensional feature spaces. The increase of feature space is due to the existence of all network features which affect the detection rate, because of measuring the distance between irrelevant service-based features. Our OCC model divide the feature space based on the service used and detect attacks based on the standard deviation of service's normal behavior which means that we use only the relevant feature space of that service, hence reducing the high dimensional network feature spaces and also ensure that each class has - a proximately - uniform distribution.

Winter, Hermann et al. [25] proposed inductive network intrusion detection system. The system operates on lightweight network flows and uses One-Class Support Vector Machines for analysis. But the system is trained with malicious rather than with normal network data. Evaluations brought satisfying results. They achieved 0% false alarm with detection rate around 98%. The drawbacks of this work that the attack variations are unlimited, this leads to have big differences in class density which affect the detection performance of the OCSVM. Also it is impossible to have a representative dataset of all possible attacks that could happen in the future. Our primary model learns the service's normal behavior instead of learning the attacks' behaviors. which means that we don't need a representative dataset of all possible attacks to build our model.

Giacinto, Perdisci et al. [26] proposed an unlabeled Network Anomaly IDS based on a modular Multiple Classifier System (MCS). Each module is designed to model a particular group of similar protocols or network services. The use of a modular MCS allows the designer to choose a different model and decision threshold for different (groups of) network services. This also allows the designer to tune the false alarm rate and detection rate produced by each module to optimize the overall performance of the ensemble. Experimental results on the KDD Cup'99 dataset [27] show that the proposed anomaly IDS achieves high attack detection rate and low false alarm rate at the same time. They achieve detection rate around 94% with false alarm around 9%. Their work is similar to ours but differs in the technique used. They use v-SVM to build their OCC model. Beside the advantages of SVMs, they have an important practical problem that is not entirely solved, which is the selection of the kernel function parameters - for Gaussian kernels the width parameter σ [29, 30]. Our OCC-NIDS model uses the standard deviation which is obtained from the data itself.

Ma and Dai [31] proposed anomaly detection using dissimilarity-based one-class classifiers (DBOCCs) with unsupervised learning approach. Several combinations of DBOCCs scheme have also been used. This technique is proposed in order to solve the drawback of traditional features-based classifiers which suffer from the improper features selection. The dissimilarity based OCCs are constructed on

dissimilarity representations (DR). The experimental results on KDD Cup'99 [27] dataset show that DBOCCs can achieve high detection rate and low false positive rate without large degeneration in performance, as traditional feature-based classifiers suffered when different feature subsets have been used. They achieve 95% detection rate with individual OCC, and around 98% with combined OCC. They didn't show the false alarm rate. The proposed algorithm suffers from high dimensional feature spaces. The increase of feature space is due to the existence of all network features which affect the detection rate, because of measuring the distance between irrelevant service-based features. Our primary model reduces the number of features for each service, that each service has its own features space.

Zainal, Maarof et al. [32] proposed an ensemble of one-class classifiers where each adopts different learning paradigms. The techniques deployed in this ensemble model were; Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF). The strengths from the individual models were evaluated and ensemble rule was formulated. Prior to classification, a 2-tier feature selection process was performed to expedite the detection process. The feature set is selected for each attack type, DoS, Prope, R2L and U2R, in addition to the normal class, and the output is one of the five classes. Empirical results on KDD Cup'99 dataset [27] show an improvement in detection accuracy for all classes of network traffic; except DoS and U2R with 97.43% and 88% respectively. The overall accuracy of their model is 96.57%. The shortcomings of their approach is the cost to classify a new instance which need to be passed through the three OCC to make decision about its class, in addition, U2R attacks are service dependent attacks which means that their features set varies based on the service that they exploit. Our OCC-NIDS model divide the feature space based on the service used and detect attacks based on the standard deviation of service's normal behavior which means that the detection of U2R attacks may achieve high detection rate because of looking at the attack from same context.

III. METHODOLOGY

All We proposed OCC-NIDS model in detecting network intrusions based on two assumptions, the first assumption is 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" [11]. This assumption is essentially in any learning methodology. The second assumption is that "The attack traffic is statistically different from normal traffic" [33, 34]. This model is based mainly on the existence of enough service's normal behaviour data. But extracting normal activities from 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 used the proposed method in [11] by which we will be highly guaranteed that the normal activity is relatively free from any type of attacks.

To overcome the existence of intrusions in the normal data, we follow the following steps which is done with each service's normal instances:

(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 [35] to detect the abnormal instances in the normal dataset and eliminate all instances that have an outlier probability greater than 0.7.

(3) After that we calculated the standard deviation for each of service class. The standard deviation of the normal class is used as the class radius or the class boundary and any new instance which have a distance from the class centroid greater than the standard deviation of that class, it is labeled as abnormal. The standard deviation is used to eliminate any abnormal behavior in the normal class and gives us the normal behavior boundaries.

As illustrated in Fig. 1, there are 3 classes, the class with circle instances, which is the biggest scattered one, is the normal class and the others, with square and triangle instances, are the abnormal classes. The normal boundary from the class 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 class 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 tune value, which is used as a standard deviation expander which added to the standard deviation's of the normal class in order to expand the class boundaries.

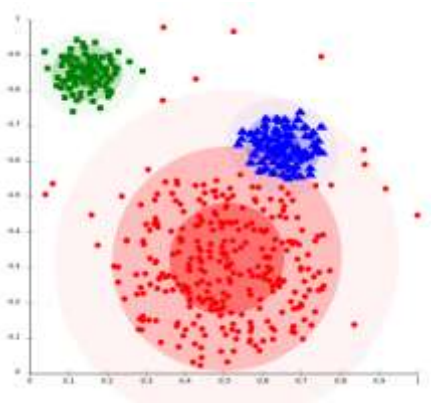

Fig. 1 The circle instances cluster is the normal behavior with standard deviation from center to the first inner circle [11]..

IV. THE PROPOSED MODEL

OCC-NIDS consists of two main phases, training phase shown in Fig. 2 and testing phase shown in Fig. 3. In training phase, training normal dataset of a certain service is extracted. After that, stratified sampling and outlier detection are performed in order to eliminate infrequent instances. Converting categorical features into binary features then applying z-score normalization to all features in order to standardize their values around zero using Eq. 1

$$vi' = \frac{vi - \mu A}{\sigma A} \quad \text{(Eq. 1) [36]}$$

After service's normal class preprocessing, generate its centroid table.

Service feature selection process is done using brute force method by choosing a subset of feature, then calculate the standard deviation on the selected features of the normal class instances using Eq.2. and Eq.3. In Eq.2 Euclidean distance is used in order to measure the distance of an instance from the service's normal class centroid table.

$$Xdist = \sqrt[2]{\sum_{i=1}^{F}(Xi - Ci)^2} \quad \text{(Eq. 2) [36]}$$

After that the sample standard deviation of service's normal class is obtained by applying Eq.3, where all normal class instances' distances is squared then summed after that divided by the total number of normal instances minus 1 then the square root of them is the sample standard deviation of the service's normal class.

$$S = \sqrt[2]{\frac{1}{N-1}\sum_{i=1}^{N}(Xdist_i} \quad \text{(Eq. 3) [11]}$$

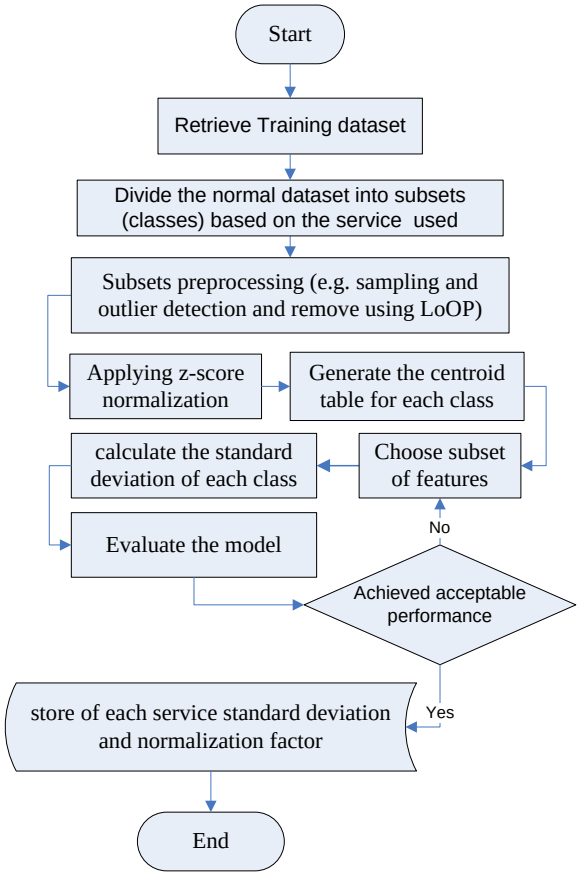

Fig. 2 OCC-NIDS Training Phase

The model evaluation process is done by measuring the classification correctness as described in the methodology step 3 . The training process loop until achieved the best features subset, where the standard deviation using the optimal features subset is stored to be used in the testing phase, also the normalization factor is also stored. Note that in the training phase we use the attack just for measuring the performance of the selected features set.

In the testing phase, which shown in Fig.3, the testing dataset, which contains both normal and attack instances, is used. In this phase the dataset features are normalized based on the normalization factor of service's normal class which is used to build the model. The test is done by retrieve each instance from the testing dataset and calculate its distance from the service normal class's centroid table, after that comparing this distance with the service normal class's standard deviation. If the instance's distance is greater than the standard deviation then label it as abnormal else label it as normal. After all testing dataset's instances are processed, calculate the overall accuracy.

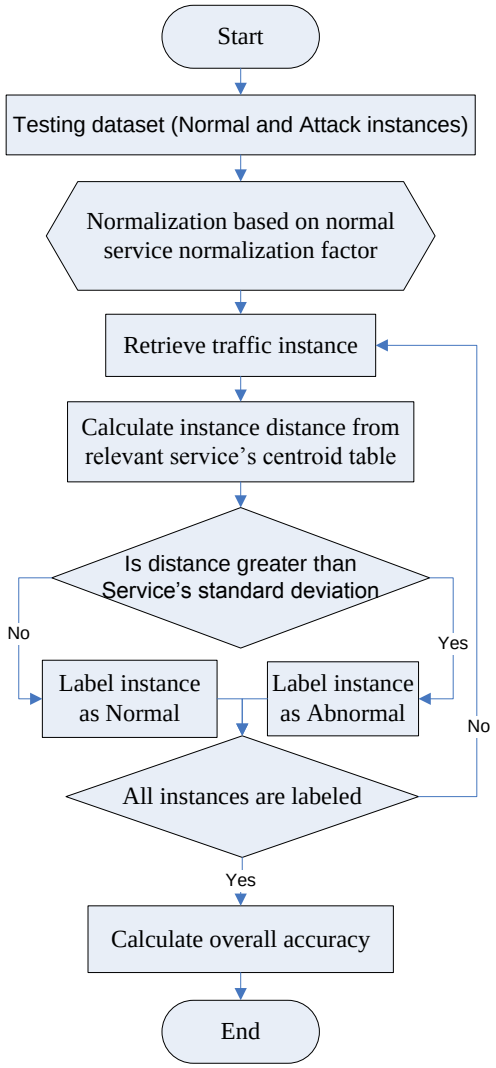

Fig. 3 OCC-NIDS Testing Phase.

V. Dataset

We evaluated our proposed model using network real data known as KDD Cup 1999 dataset [27] which was prepared and managed by MIT Lincoln Labs. This dataset is used as a benchmark for intrusion detection systems, and it is widely used and accepted in the academic community. The training data includes 22 different attacks out of the 39 present in the test data. In the testing dataset there are novel attacks that do not exists in the training dataset. 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. The training dataset consists 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.

VI. Experimental Setup

For building the proposed OCC-NIDS model we chose HTTP service as the service by which we need to evaluate OCC-NIDS model, note that each service needs its own OCC. We chose HTTP service because of the existence of enough normal instances in addition to the existence of varying attack types as shown in Table 1.

Table 1
HTTP SERVICE TRAINING DATASET

10% training dataset has 61,886 normal instances			
Attack	**Original#**	**Total**	**Type**
satan.	2	1416	Probe
neptune.	192	107,201	DoS
portsweep.	3	1039	Probe
phf.	4	4	R2L
ipsweep.	3	94	Probe
back.	2203	2203	DoS

Note that we included all the probe attacks and neptune attack that exploited TCP protocol into HTTP because the original number of attacks are very small which don't represent the attack behaviour and don't give a true indication about our OCC-NIDS model performance.

Table 2
HTTP SERVICE TESTING DATASET

10% testing dataset has 39,247 normal instances			
Attack	**Original#**	**Total**	**Type**
apache2.	794	794	DoS
neptune.	93	58001	DoS
portsweep.	2	354	Prope
phf.	2	2	R2L
saint.	1	607	Prope
back.	1098	1098	DoS

Table 2 listed the attacks that exploited the HTTP service in the testing dataset, we also included all the probe attacks and neptune attack that exploited TCP protocol into HTTP because the original number of attacks are very small which don't represent the attack behaviour and don't give a true indication about our OCC-NIDS model performance.

By applying training phase, shown in Fig.2 on training dataset shown in Table 1, we obtained the optimal features set of the HTTP service which are listed in Table 3. The model results using these features set is shown in Table 4 and Fig.4. As shown, the model achieved, using the optimal features set, 100% detection rate of attack instances while 97.7 for normal instances at a tune value equals 6. As we increase the tune value, the normal detection rate in

increased whereas the attack detection rate is decreased. Note that without expanding value of the standard deviation (Tune value=0) 83.9% of normal instances have been correctly classified as normal.

TABLE 3
THE SELECTED FEATURES OF HTTP SERVICE

#	Feature Name	#	Feature Name
1	Flag	11	srv_rerror_rate
2	logged_in	12	same_srv_rate
3	src_bytes	13	srv_diff_host_rate
4	Hot	14	diff_srv_rate
5	num_compromised	15	dst_host_count
6	count_v	16	dst_host_srv_count
7	serror_rate	17	dst_host_diff_srv_rat
8	srv_serror_rate	18	dst_host_same_src_p

TABLE 4
OCC-NIDS TRAINING PHASE RESULTS USING OPTIMAL FEATURE SET

Label	Tune Value					
	0	6	7	10	15	25
normal.	0.839	**0.977**	0.986	0.994	0.996	0.997
back.	1.000	**1.000**	1.000	1.000	1.000	1.000
ipsweep.	1.000	**1.000**	0.979	0.968	0.968	0.968
neptune.	1.000	**1.000**	1.000	1.000	1.000	1.000
phf.	1.000	**1.000**	1.000	1.000	1.000	0.000
portsweep.	1.000	**1.000**	0.999	0.997	0.993	0.983
satan.	1.000	**1.000**	1.000	1.000	1.000	1.000

We observed that the detection rate of DoS attacks, back and neptune attacks, is 100% at a tune value equals 25 with false alarm equals 0.3%. Whereas the detection rate of R2L attack called phf become 0% but the probe attacks, ipsweep and portsweep have a detection rate higher than 96.8% at tune value equals 25. We need to test the detection ability of our OCC-NIDS model using the testing dataset at tune value equals 6.

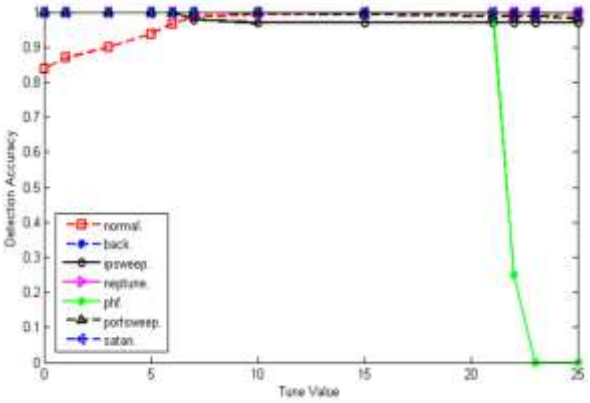

Fig. 4 OCC-NIDS training phase chart using optimal feature set

VII. OCC-NIDS MODEL RESULTS

The results of the testing phase is listed in Table 5. As shown, at tune value 6 we achieved 99.3% detection rate of normal instances and 100% detection rate of all attack classes except a new DoS attack type called apache2 detected with rate equals 99.6%. Another new attack was detected with 100% detection rate, this attack is a probe attack called saint attack.

TABLE 5
OCC-NIDS TESTING PHASE RESULTS USING OPTIMAL FEATURE SET

Label	Tune Value					
	0	6	7	16	23	25
normal.	0.916	**0.993**	0.993	0.996	0.997	0.997
apache2.	0.999	**0.996**	0.996	0.984	0.976	0.975
back.	1.000	**1.000**	1.000	1.000	1.000	1.000
neptune.	1.000	**1.000**	1.000	1.000	1.000	1.000
phf.	1.000	**1.000**	0.500	0.500	0.000	0.000
portsweep.	1.000	**1.000**	1.000	0.969	0.958	0.958
saint.	1.000	**1.000**	1.000	1.000	1.000	1.000

As shown in Table 5 and Fig. 5, the detection rate of phf attack is decreased to 50% directly at tune value7. As we increase the tune value, the detection rate of normal instances is increased and the detection rate of attack instances is decreased. The detection rate of DoS attacks didn't decreased despite we increased the tune value reach 25. This means that DoS attacks are far a way from the services class's standard deviation.

Phf attack is an R2L attack, which is a dangerous attack, because it violates system integrity and confidentiality. So this type is needed to be detected even with relatively high false alarm. Phf attack was detected with 0.7 false alarm, which is an acceptable false alarm rate. Table 6 presents the confusion matrix results in addition the attack detection rates in addition to the number of true negative instances and the number of false positive instances. As shown in Table 6, the model achieved an overall detection rate equals 99.72% and accuracy rate equals 99.65% with false positive rate equals 0.005% and false alarm rate equals 0.7%.

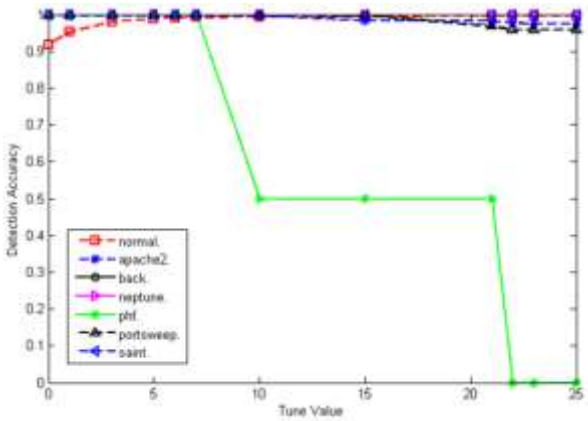

Fig. 5 OCC-NIDS testing phase chart using optimal feature set

TABLE 6

OCC-NIDS TESTING PHASE CONFUSION MATRIX RESULTS

TP%	FP%	Detection%	Accuracy	Correl.	F-1
99.29	0.005	99.720	99.645	0.993	0.996

Class	Attack	TN#	FP#	TN%	FP%
DoS	neptune.	58001	0	100	0
	apache2.	791	3	99.62	0.379
	back.	1098	0	100	0
Probe	saint.	607	0	100	0
	portsweep.	354	0	100	0
R2L	phf.	2	0	100	0

VIII. DISCUSSION

OCC-NIDS achieved high detection rate, low false positive rate and low false alarm rate. The overall detection rate achieved was 99.72% with accuracy reached 99.65% and false positive rate equals 0.005% and with false alarm equals 0.7%. although the varying distribution of testing dataset attacks [27], the model prove its ability of detecting even new attacks which are a DoS attack called apache2 and a probe attack called saint with a detection rates 99.62% and 100% respectively.

We observed from the results of training phase and testing phase that our OCC-NIDS was very robust against DoS attacks and perform well with Probe attacks although that the Probe attacks are a stealthy attacks which are hard to be detected. We don't sure if our model is robust against R2L attacks or weak because there's no representative instances of this type of attacks. In the training dataset where are just four instances for phf attack and in the testing dataset exists just two instances of the same attack.

IX. CONCLUSION

We proposed OCC-NIDS model based on the standard deviation of service's normal behaviour. Through this model we dealt with each network service as single class instead of dealing with all network services as a single class. By this way we use just the relevant features of each service, hence reducing the high dimensional network feature spaces and also ensure that each class has - a proximately - uniform distribution.

The proposed OCC-NIDS model has the advantage of detecting attacks without a previous knowledge about their behaviour. The model just learn the normal behaviour. The standard deviation with a tune value is used to determine the normal class's boundaries and any instance has a distance greater than the standard deviation of that normal class is labeled as abnormal. The results show that our model has the ability to detect new attacks with high detection rate and low false alarm rate, especially DoS attacks.

X. FUTURE WORK

In our future work, we'll focus on extracting the most relevant service's features based on the normal instances by considering the variance of each feature.

REFERENCES

[1] ISC, *ISC Internet domain survey (January 2015).* Internet Systems Consortium, Inc . http://ftp.isc.org/www/survey/reports/2015/01. Accessd on: 13/02/2015, 2015.

[2] Ben-David, S., T. Lu, and D. Pál. *Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning*. in *COLT*. 2008.

[3] McHugh, J., *Intrusion and intrusion detection.* International Journal of Information Security, 2001. **1**(1): p. 14-35.

[4] Moore, D., et al., *Inside the slammer worm.* IEEE Security & Privacy, 2003. **1**(4): p. 33-39.

[5] Heady R, et al., *The Architecture of a Network Level Network Intrusion Detection System. (Technical Report CS90-20)* University of New Mexico: Department of Computer Science, 1990.

[6] Sobh, T.S., *Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art.* Computer Standards & Interfaces, 2006. **28**: p. 670–694.

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

[8] 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.

[9] 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.

[10] 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.

[11] Barhoom, T.S. and R.A. Matar, *Network Intrusion Detection Using Semi-Supervised Learning Based on Normal Behaviour's Standard Deviation.* Network, 2015. **4**(1).

[12] Khan, S.S. and M.G. Madden, *A survey of recent trends in one class classification*, in *Artificial Intelligence and Cognitive Science*2010, Springer. p. 188-197.

[13] Laskov, P., et al., *Learning intrusion detection: supervised or unsupervised?*, in *Image Analysis and Processing–ICIAP 2005*2005, Springer. p. 50-57.

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

[15] 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.

[16] 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.

[17] Li, J., W. Zhang, and K. Li, *A Novel Semi-supervised SVM based on Tri-training for Intrusition Detection.* Journal of computers, 2010. **5**(4): p. 638-645.

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

[19] Rassam, M.A., M. Maarof, and A. Zainal, *A survey of intrusion detection schemes in wireless sensor networks.* American Journal of Applied Sciences, 2012. **9**(10): p. 1636-1652.

[20. Zhu, X., *Semi-supervised learning literature survey.* Computer Sciences Technical Report 1530, University of Wisconsin–Madison, 2005.
[21] Wang, J., K. Zhang, and D.-s. Ren. *An anomaly intrusion detection algorithm based on minimal diversity semi-supervised clustering*. in *Computer Science and Computational Technology, 2008. ISCSCT'08. International Symposium on*. 2008. IEEE.
[22] Lu, T.T., *Fundamental limitations of semi-supervised learning.* M.S. thesis, Dept. of Comput. Sci., Univ. of Waterloo, Waterloo, ON, Canada, 2009.
[23] Li, K.-L., et al. *Improving one-class SVM for anomaly detection*. in *Machine Learning and Cybernetics, 2003 International Conference on*. 2003. IEEE.
[24] Araki, S., et al. *Unknown Attack Detection by Multistage One-Class SVM Focusing on Communication Interval*. in *Neural Information Processing*. 2014. Springer.
[25] Winter, P., E. Hermann, and M. Zeilinger. *Inductive intrusion detection in flow-based network data using one-class support vector machines.* in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. 2011. IEEE.
[26] Giacinto, G., et al., *Intrusion detection in computer networks by a modular ensemble of one-class classifiers.* Information Fusion, 2008. **9**(1): p. 69-82.
[27] KDD, *The third international knowledge discovery and data mining tools competition dataset (KDD99 Cup). http://kdd.ics.uci.edu/databases/kddcup99/ ; Accessed on: 24/12/2014.* 1999.
[28] Kyoto2006+, *Dataset, http://www.takakura.com/Kyoto_data/*. 2009.
[29]. Suykens, J.A., *Advances in learning theory: methods, models, and applications*. Vol. 190 P. 391. 2003: IOS Press.
[30] Olson, D.L. and D. Delen, *Advanced data mining techniques*2008: Springer Science & Business Media.
[31] Ma, J. and G. Dai. *Anomaly detection in computer networks using dissimilarity-based one-class classifiers*. in *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*. 2008. IEEE.
[32] Anazida Zainal, Mohd Aizaini Maarof, and Siti Mariyam Shamsuddin, *Ensemble Classifiers for Network Intrusion Detection System.* Journal of Information Assurance and Security, 2009. **Vol. 4 p. 217-225**.
[33] Javitz, H.S.V., A., *The NIDES statistical component: Description and justication.* Technical report, SRI International., 1993.
[34] Denning, D., *An intrusion detection model.* In IEEE Transactions on Software Engineering 13., 1987.
[35] 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.
[36] Jiawei Han, Micheline Kamber, and Jian Pei, *Data mining :s concepts and techniques, 3rd ed.*2012, 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers is an imprint of Elsevier.