Evaluation of K-Means Clustering for Effective Intrusion Detection and Prevention in Massive Network Traffic Data

Kamini Nalavade
Research Scholar,
Computer Department, VJTI, Matunga
Mumbai, India

B. B. Mehsram, Ph.D
Professor and Head, Computer Department,
VJTI, Matunga
Mumbai, India

ABSTRACT
With the growth of hacking and exploiting tools and invention of new ways of intrusion, Intrusion detection and prevention is becoming the major challenge in the world of network security. It is becoming more demanding due to increasing network traffic and data on Internet. There are various approaches being utilized in intrusion detections, but unfortunately any of the systems so far is not completely flawless. So, the quest of betterment continues. Intrusion detection systems using data mining approaches make it possible to search patterns and rules in large amount of audit data. Classification-based data mining models for intrusion detection are often ineffective in dealing with dynamic changes in intrusion patterns and characteristics. Unsupervised learning methods are efficient in detecting unknown attacks in large datasets. In this paper we investigate clustering approaches for network intrusion detection. We carried out our experiments on K-means clustering algorithm and measured the performance based on detection rates and false positive rate with different cluster values. The KDD dataset which is freely available online is used for our experimentation and results are compared. Our intrusion detection system using clustering approach is able to detect different types of intrusions, while maintaining a low false positive rate.

General Terms
Security, Intrusion Detection, Data Mining

Keywords
Network, Attacks, k-means Clustering, Security

1. INTRODUCTION
Communication on Internet, in spite of implementation of advanced security measures, is always under innovative and inventive attacks. Given the different type of attacks like Denial of Service, Spoofing, Session hijacking, password guessing and others, it is a challenge for any intrusion prevention system to detect a wide variety of attacks. The goal of intrusion prevention systems is to automatically detect attack from the stream of network audit trails. Once an attack is detected, alarm is generated for administrator and action against the intrusion is taken. Generally signature based systems are used to detect known attacks. These methods are provided with signatures of attacks and perform rule matching to detect intrusions. But these kinds of systems are not sufficient to detect new or unknown attacks. In such cases use of data mining and machine learning approaches can be used for intrusion detection.

Classification approaches demand training dataset or labelled data set which is practically very difficult to generate. As we deal with large volume of network data, it is difficult to label each instance manually or classify each record manually. We need an approach which can classify the records when training data set is unlabelled. Unsupervised learning methods assume that training data is not labelled. Clustering algorithms have gained consideration as they can support present intrusion detection and prevention systems in several respects. A significant advantage of using clustering or unsupervised learning to detect network attacks is the ability to find new attacks or zero day attacks. This indicates that attack types with unknown pattern signatures can be detected using this approach. Clustering results can also assist the network security administrator with labelling network traffic records as normal or intrusive. The amount of available network traffic audit data is usually large, making the labelling process of all records very time-consuming, and costly. Additionally, labelling a large number of network traffic records can lead to errors being fused during the process. Grouping similar data together eases the task of labelling. Our approach is to build a probabilistic model from the training data and then using them to determine whether a new record is anomaly or not.

The most important objective of this paper is to evaluate performance of k-means algorithm for design of clustering based intrusion prevention system. Therefor compare the performance of k-means algorithm with different values of clusters. We investigated the performance of k-means clustering algorithm in terms of performance criteria as detection rate and false positive rate. This paper demonstrates that our clustering based method can outperform and enhance intrusion detection results in detecting unseen attacks. The paper is organised as follows: Section 2 depicts the related work. In section 3 we explain the methodology for performance measurement of k-means algorithm for intrusion detection. In section 4 we describe our experimentation and results in two parts. First is the comparative analysis of K-means, algorithm with different sets of data. Second we describe performance analysis of K-means algorithm with different cluster values. In last section we conclude the paper with future scope.

2. RELATED WORK
Unsupervised anomaly detection techniques detect anomalies in an unlabelled data set assuming that the maximum numbers of instances in the dataset are normal and do not require training data. The techniques in this category make the implicit assumption that normal instances are more frequent than anomalies in the test data. If this assumption is not true then such methods suffer from high false alarm rate. Such deviation assumes that the test data contains very few anomalies and the model learned during training is robust to these few anomalies.

Clustering is quite a topic of interest for research over many years. Related work on clustering-based intrusion detection focus on constructing a set of clusters based on unlabelled [11, 13] or labelled [12] training data to classify test data records. Clustering methods include Linkage based and k-means techniques. Some other techniques for clustering include density based methods such as Dbscan, AI based methods such as Self-Organizing Maps and growing networks. Lee[8] emphasized the data-flow environment of network intrusion detection, targeting at real-time feature extraction and classification from network traffic data. Potnoy[11] presented a clustering method for detecting intrusions from unlabelled data. Unlike traditional anomaly detection methods, they cluster data records that contain both normal behaviours and attacks, using an incremental k-means algorithm. After clustering, each cluster is labelled based on the number of instances in the cluster. The investigations show that very small clusters tend to be attacks. The self-labelled clusters are then used to detect attacks from test dataset. Ye and Li[12] proposed a supervised clustering technique that constructs clusters from labelled training data and uses them to score the possibility of being attacks for test data instances. Performance better than decision tree classification models was reported. Guan [13] detected intrusions with a different clustering algorithm, namely an improved k-means algorithm that addresses the selection of number of clusters and the elimination of empty clusters. M,Varaprsad Rao[26] used the k-Means clustering algorithm to partition a dataset into meaningful patterns. Modified k-Means is applied in pre-processing and normalization steps. As a result the effectiveness is improved and it overcomes the flaws of k-Means. This approach is proposed to work on network intrusion data and the algorithm is experimented with KDD99 dataset and found satisfactory results.

Zhenglie Li[27] proposed k-means clustering algorithm is an effective method to the intrusion detection system. Particle swarm optimization (PSO) algorithm which is evolutionary computation technology based on swarm intelligence has good global search ability. Experiments on data sets KDD CUP 99 has shown the effectiveness of the proposed method and also shows the method has higher detection rate and lower false detection rate. Hamdan [30] explained the process of intrusion detection which is the major part of network activity and security policies adapted over the network to secure it. In this research paper four intrusion detection approaches, Artificial Neural Network,

Chandola et. al.[31] refers to the problem of finding patterns in data that do not conform to expected behaviour of network traffic. These non-conforming patterns are often referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains. Ahmed [33] proposed machine learning approach in detecting the anomalies in the network. In this research paper it is explained that Machine learning techniques enables the development of anomaly detection algorithms that are non-parametric, adaptive to changes in the characteristics of normal behaviour in the relevant network, and portable across applications. They investigated the use of the block-based One-Class Neighbour Machine and the recursive Kernel-based Online Anomaly Detection Algorithm. Jhang[34] presents an survey on anomaly detection. The authors have included test and training both types of data. In order to distinguish between the different approaches used for anomaly detection in networks, they have classified those methods into four categories: statistical anomaly detection, classifier based anomaly detection, anomaly detection using machine learning and finite state machine anomaly detection. Each method is described in detail along with examples for its applications.

3. METHODOLOGY

In this section we describe our methodology for detecting intrusions using K-means clustering algorithm. We first describe the dataset and how it is used to build clusters for intrusion detection.

3.1 Dataset and Normalization of data

The dataset used is KDDcup1999 intrusion dataset which contains wide variety of intrusions simulated in network environment to acquire nine weeks of raw TCP dump data for a local-area network. Since 1999, KDD'99 [3] has been the most wildly used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. [2] and is built based on the data captured in DARPA'98 IDS evaluation program. A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address. Each connection is labelled as either normal, or as an attack, with exactly one specific attack type. It is important to note that the testing data is not from the same probability distribution as the training data. This makes the task more realistic. The datasets contains a total of 22 training attack types. There are 41 features for each connection record that are divided into discrete sets and continuous sets according to the feature values. It consists of number of total records 494021. The 22 different types of network attacks in the KDD99 dataset fall into four main categories:

1) Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate re-quests, or denies legitimate users access to a machine.

2) User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

3) Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

4) Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

The attacks in each class are as shown below:

Table 1: Classes of Attacks

S.N.	Class	Attack Types
1	DOS	Back, Land, Neptune,pod, smurf, Teardrop,
2	U2R	Buffer_overflow, loadmodule, perl, rootkit
3	R2L	ftp_write, guess_passwd, imap, multihop, phf, spy,warezlient, warezmaster
4	Probe	IPsweep,nmap, satan,portsweep

Normalization of dataset

As we are interested in developing a general system for intrusion detection, it must be able to create clusters from an arbitrary distribution. If the training set and test set are from different distribution and cluster width is fixed, then the

clustering may result in improper placement of records. There are two solutions to this problem. One is to use dynamic cluster width depending upon the distribution. Second one is to convert the test set according to the distribution of training set assuming that training set accurately reflects the range and deviation of feature values. When given a training set we can calculate mean and standard deviation feature vectors using following:

$$meanvector[j] = \frac{1}{N}\sum_{i=1}^{N} instance_i[j]$$

The standard deviation will be calculated as follows

$$stdvector[j] = \left(\frac{1}{N-1}\sum_{i=1}^{N} (instance_i[j] - meanvector[j])^2\right)^{1/2}$$

Every record is converted to new record by using the following

$$newinstance[j] = \frac{instance[j] - meanvetor[j]}{stdvector[j]}$$

For every instance value we calculate deviation between feature value and average vector. But the limitation is, we can apply this to only continuous feature values.

3.2 Distance Metric

For detecting intrusions some feature values are more useful than others or we can say they are highly weighted than others. Difference in the values of such feature also has more importance. To calculate difference in values we have used Euclidean distance as metric for equally weighted features. Also normalization process converts the data set into standard form. The Euclidean method added a constant value to the squared distance between two instances for every discrete feature with different values.

3.3 Clustering

To create clusters from the input data, we have used k-means clustering algorithm. K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The algorithm initially have empty set of clusters and updates it as proceeds. For each record it computes the Euclidean distance between it and each of the centroids of the clusters. The instance is placed in the cluster from which it have shortest distance. Assume we have fixed metric M, and constant cluster Width W. Let di(C, d) is the distance with metric M, Cluster centroid C and instance d where centroid of cluster is the instance from feature vector.

Input: The number of clusters K and a dataset for intrusion detection
Output: A set of K-clusters

Algorithm:

1. Initialize Set of clusters S. (randomly select k elements from the data)

2. While cluster structure changes, repeat from 2.

3. Determine the cluster to which source data belongs Use Euclidean distance formula.

Select d from training set. If S is empty, then create a cluster with centroid as d.

else add d to cluster C with min (dist (C, d)) or

dist(C ,d)<=dist(C1, d).

4. Calculate the means of the clusters. Change cluster centroids to means obtained using Step 3.

3.4 Cluster Classification

If cluster width is chosen properly then after clustering each cluster contains instance of same type. The major task is to determine which clusters are normal and intrusive in case of intrusion detection. Here we assume that maximum numbers of records are normal from the training set. Then it is highly possible that the cluster with maximum numbers of instances contains normal records and other contains attack records. We have used 75% as threshold percentage value for labelling the normal cluster. The other clusters are labelled as anomalous.

3.5 Intrusion Detection

Clustering creates clusters of normal and anomalous instances. The next task is to perform intrusion detection from the test set. Suppose d is the instance from the test set, we perform intrusion detection as follows

1. Normalize d based on the statistical information from the training set, let it be d'.

2. Find a cluster closest to d' using the metric Euclidean distance.

3. Suppose dist(C,d')<= dist(C1, d'), then place the instance in cluster C.

4. Classify the record as cluster label of C (normal or anomalous).

In other words, we classify the test instance based on the shortest distance from the cluster centroids. As the normalization is performed using mean and standard deviation, the chances of selection of wrong cluster are reduced. We agree that an online, real-time, and adaptive intrusion detection system is the goal of this research, towards which our clustering-based approaches have provided a promising tool. The purpose of unsupervised intrusion detection is to discover new attacks in a new dataset. It is more practical to run clustering algorithms on the new dataset and identify attacks by self-labelling.

4. EXPERIMENTATION & RESULTS

In this section we demonstrate how the system parameters are decided and fixed. Also the different performance parameters used to evaluate the system. Then we discuss the results achieved in intrusion detection using k-means clustering algorithm on KDD cup dataset. Next we evaluated the performance of k-means clustering algorithm with different values of initial cluster.

4.1 Performance Parameters

There are many measures available for evaluating system performance. For evaluating intrusion detection results following measure are generally used.

1. True positive (TP) means number connections that were correctly classified as intrusion.

2. True Negative (TN) means number of connections that were incorrectly classified as intrusion.

3. False positive (FP) means number of intrusion connections that were incorrectly classified as normal.

4. False negative FN) means number of normal connections that were incorrectly classified as intrusion.

To determine how many misclassification are found we use term Recall. Precision is how many records are correctly classified by the system.

Precision=TP/TP+FP……..(1)

Recall=TP/TP+FN…… .(2)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{.....(3)}$$

Confusion Matrix for Intrusion Protection System

Table 2: Confusion Matrix

	Predicted Class		
Actual Class		Normal	Attack
	Normal	TP	FN
	Attack	FP	TN

We mainly concentrate on false positive rate (fpr), recall, precision and overall accuracy. In the next two sections, we present two sets of experiments, each designed to demonstrate a different point. The first set is used to intrusion detection results using unsupervised anomaly detection method. We also show that clustering methods can be employed to help the performance of classification-based intrusion detection techniques. The second set of results shows that how the different cluster values for k-means algorithm affects the performance of system.

4.2 Experimental Results-I

For our intrusion prevention system we are mainly interested in false positive rate, recall, precision and accuracy. These are the best measures for evaluating system performance as they determine percentage of intrusions detected as well as incorrect classification done by the system. There are two important parameters are required to be fixed. First is the cluster width and second is threshold percentage value for normal clusters. Cluster width determines minimum distance between two instances. Threshold percentage for normal cluster is required as we have assumed that maximum instances are normal and classified in one cluster. We experimented on 10% KDD dataset to fix these values. We used same set as training and testing purpose. To determine the threshold percentage we kept fixed cluster width and to determine the cluster width we experimented with different values cluster width and checked accuracy and FPR.

Table 3: Threshold Percentage Measurement

Threshold percentage	False Positive rate
4 %	6.482%
11%	4.352%
14%	2.647%
18%	1.746%

For the above results the cluster width was fixed to 25. As we can observe we acquired lowest false positive rate for the threshold value 18%. So we fix this value for our study.

Table 4: Cluster Width Measurement

Cluster Width	False Positive rate	Detection rate
10	2.843%	26.38%
15	1.752%	28.49%
30	1.019%	28.05%
40	0.76%	31.62%

The Cluster width is fixed to 30 after the above results are observed. The next step is to proceed with the clustering approach. We partitioned the KDD dataset into 10 subsets each containing 4,90,000 records. After classification we observed that many subsets contain single type of instances such as only smurf records or backdrop records. Requirement for the system is that it must contain all types of values in the set. Otherwise using these training set may fail to create clusters for other records. So out of these 10 subsets five subsets were selected which contain all types of instances. Since our aim is to detect network intrusion using clustering algorithms, we now analyze the unsupervised intrusion detection accuracies. For the next measurement of k-means clustering algorithm, we decided the no. of clusters to be 3. The evaluation of K-means with the five subsets of KDD dataset is shown below in the table.

Table 5: Performance of Intrusion Detection System

Set	TP	FP	TN	FN	Acc	Prec	Recall
1	0.803	0.046	0.625	0.086	0.9153	0.9458	0.9032
2	0.992	0.034	0.953	0.006	0.9798	0.9668	0.9939
3	0.953	0.013	0.975	0.041	0.9727	0.9865	0.9587
4	0.931	0.021	0.978	0.013	0.9825	0.9779	0.9862
5	0.965	0.019	0.988	0.023	0.9789	0.9806	0.98

We now discuss the accuracy results at one cutting point from the sorted list of clusters. Suppose the approximate percentage of attack instances is known apriori or from heuristics, we split the sorted cluster list at a point that generates the desired percentage. In this paper, we group the clusters as normal or intrusive in such a way that the number of data instances in attack clusters account for about 18% of the total population, reflecting the assumed distribution of the training data. We run each experiment 5 times and report the overall accuracy, false positive rate, and attack detection rate. The k-means algorithm performs better than others, generating low FPR values and high overall accuracies; the performance difference to other algorithms is significant.

4.3 Experimental Results-II

We examined intrusion detection capability of k-means clustering algorithm with the above experimentation. The major parameter to be decided in the k-means clustering is the number of clusters. In the above experiments we assumed three clusters for the algorithm. But deciding precise number of clusters is again an issue for debate. Therefor we carried out experiment on different cluster values for k-means. We examined our results in three rounds in decreasing size of clusters. We applied K-means clustering algorithm. Initially K is 5, then K is 4 and finally K is 2.

Table 6: Clustering K-means Algorithm

Attacks	K=5		K=4		K=2	
	Rec	Preci	Recal	Preci	Reca	Preci
Normal	0.74	0.73	0.99	0.71	0.99	0.63
DOS	0.50	0.96	0.71	0.96	0.78	0.98
Probe	0.56	0.04	0.64	0.67	0.71	0.70
R2L	0.59	0.41	0.00	0.01	0.00	0.01

There is no general theoretical solution to find the optimal number of clusters for any given dataset. We choose K=2 for the experimentation. K-means algorithm achieves the high quality in recall and precision with much less run time. Comparing the results, we observe that the k-means algorithm scale linearly with the number of clusters. Although the batch k-means is computationally efficient, it does not generate coherent clusters like the other algorithms. The k-means algorithm seems to be a desirable choice, with high clustering quality and relatively low time complexity.

5. CONCLUSION & FUTURE WORK

The feasibility of unsupervised intrusion detection using clustering algorithms is investigated in this study. Considering the vibrant nature of network traffic intrusions, unsupervised intrusion detection is more suitable for anomaly detection than classification-based intrusion detection methods. Our experiment shows that k-means algorithm achieves a very good performance with more than 90% accuracy in intrusion detection. Our experiment about deciding the minimum distance between instance and centroid shows that sometimes high distance may result in more detection rate but it may cause high deviation. Comparing the results with different cluster values, we observe that the k-means algorithm scale linearly with the number of clusters. We are now experimenting comparing detection rate and false positive rates for signature based detection systems and anomaly based systems. Our future work involves development of intrusion protection system to achieve low false positive rate and more accuracy using anomaly based detection approach.

6. REFERENCES

[1] V. Kumar, Parallel and distributed computing for cybersecurity. IEEE Distributed Systems, 2005

[2] Z.-X. Yu, J.-R. Chen and T.-Q. Zhu, A novel adaptive intrusion detection system based on data mining, in Proceedings IEEE International Conference on Machine Learning and Cybernetics(2005), pp. 2390–2395.

[3] X. Zhu, Z. Huang and H. Zhou, Design of a multi-agent based intelligent intrusion detection system, in Proceedings 1st International Symposium on Pervasive Computing and Applications (Urumqi, China, 2006) (IEEE Computer Society), pp. 290–295.

[4] W.Li,K.Zhang,B.Li and B.Yang ,An efficient framework for intrusion detection based on data mining, in Proceedings 2005 ICSC Congress on Computational Intelligence Methods and Applications (IEEE Computer Society) (2005).

[5] C.-T. Lu, A. P. Boedihardjo and P. Manalwar, Exploiting efficient data mining techniques to enhance intrusion detection systems, in Proceedings IEEE International Conference on Information Reuse and Integration (Las Vegas, NV) (IEEE Computer Society, 2005), pp. 512–517.

[6] T. M. Khoshgoftaar, C. Seiffert and N. Seliya, Labeling network event records for intrusion detection in a wireless LAN, inforrmation Reuse and Integration (Waikoloa Village, HI) (IEEE Computer Society) (2006), pp. 200–206.

[7] J. Zhang and M. Zulkernine, A hybrid network intrusion detection technique using random forests, in Proceedings 1st International Conference on Availability, Reliability and Security (IEEE Computer Society) (2006), pp. 8.

[8] W. Lee, S. Stolfo and K. Mok, Mining in a data-flow environment: Experience in network intrusion detection, in Proc. 5th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining (San Diego, CA) (1999), pp. 114–124.

[9] D. E. Denning, An intrusion detection model, IEEE Trans. Software Engineering 13 (1987) 222–232.

[10] E. Eskin, Anomaly detection over noisy data using learned probability distributions, in Proc. 17th Int. Conf. Machine Learning (San Francisco, CA) (2000), pp. 255–262.

[11] L. Portnoy, E. Eskin and S. Stolfo, Intrusion detection with unlabeled data using clustering, in ACM Workshop on Data Mining Applied to Security (Philadelphia, PA) (2001).

[12] N. Ye and X. Li, A scalable clustering technique for intrusion signature recognition, in Proc. 2nd IEEE SMC Information Assurance Workshop (2001), pp. 1–4.

[13] Y. Guan, A. A. Ghorbani and N. Belacel, Y-means: A clustering method for intrusion detection, in Canadian Conference on Electrical and Computer Engineering, Montral, Qubec, Canada (2003), pp. 1–4.

[14] T. M. Khoshgoftaar, S. V. Nath, S. Zhong and N. Seliya, Intrusion detection in wireless networks using clustering techniques with expert analysis, in Proceedings 4t International Conference on Machine Learning and Applications (Los Angeles, CA (2005), p. 6.

[15] S. Zhong, T. M. Khoshgoftaar and N. Seliya, Evaluating clustering techniques for network intrusion detection, in 10th ISSAT Int. Conf. on Reliability and Quality Design (Las Vegas, Nevada, USA) (2004), pp. 173–177.

[16] J. MacQueen, Some methods for classification and analysis of multivariate observations. In Proc. 5th Berkeley Symp. Math. Statistics and Probability (1967), pp. 281–297.

[17] J. D. Banfield and A. E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics (1993) 803–821.

[18] T. Kohonen, Self-Organizing Map (Springer-Verlag, New York, 1997).

[19] T. M. Martinetz, S. G. Berkovich and K. J. Schulten, Neural-Gas network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Networks (1993) 558–569.

[20] B. Fischer, T. Zoller and J. M. Buhmann, Path based pairwise data clustering with application to texture segmentation, Lecture Notes in Computer Science 2134 (2001) 235–250.

[21] G. Karypis, E.-H. Han and V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, IEEE Computer (1999) 68–75.

[22] G. Karypis, CLUTO — A Clustering Toolkit , Dept. of Computer Science, University of Minnesota, May 2002. http://www-users.cs.umn.edu/ karypis/cluto/.

[23] S. Zhong and J. Ghosh, A unified framework for model-based clustering, Journal of Machine Learning Research (2003) 1001–1037.

[24] S. Zhong, T. M. Khoshgoftaar and N. Seliya, Analyzing software measurement data with clustering techniques, IEEE Intelligent Systems (2004) 20–27.

[25] SK Sharma, P Pandey, SK Tiwar "An improved network intrusion detection technique based on k-means clustering via Naïve bayes classification" IEEE Volume 2, Issue 2, February 2012, Issn 2151-961.

[26] M,Varaprsad Rao "Algorithm for Clustering with Intrusion Detection Using Modified and Hashed K – Means Algorithms "Published by IEEE Computer Society,2012

[27] Zhenglie Li"Anomaly Intrusion Detection Method Based on K-Means Clustering Algorithm with Particle Swarm Optimization "Springer Volume 4, Issue 2, April 2011.

[28] Thaksen J.Parvat" Network Log Clustering Using K-Means Algorithm'In IEEE Pasfic asia workshop of networking 2011. Dabas et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(3), March - 2013, pp. 507-511 © 2013, IJARCSSE All Rights Reserved Page | 511

[29] Asmaa Shaker Ashoor (Department computer science, Pune University) Prof. Sharad Gore (Head department statistic, Pune University), "Importance of Intrusion Detection System (IDS)", International Journal of Scientific & Engineering Research, Volume 2, Issue 1, January-2011 ISSN 2229-5518.

[30] Hamdan.O.Alanazi, Rafidah Md Noor, B.B Zaidan, A.A Zaidan, "Intrusion Detection System: Overview "Journal Of Computing, Volume 2, Issue 2, February 2010, Issn 2151-961

[31] Varun Chandola University Of Minnesota Arindam Banerjee University Of Minnesota And Vipin Kumar University Of Minnesota "Anomaly Detection : A Survey", ACM Computing Surveys, September 2009.

[32] Paul Barford University of Wisconsin, Nick Duffield AT&T, Amos Ron University and Joel Sommers Colgate, "Network Performance Anomaly Detection and Localization" Infocom 2009.

[33] Tarem Ahmed, Boris Oreshkin and Mark Coates, Department of Electrical and Computer Engineering McGill University Montreal, QC, Canada "Machine Learning Approaches to Network Anomaly Detection" in Workshop on Tackling Computer Systems Problems with Machine Learning Techniques, 2007.

[34] Weiyu Zhang; Qingbo Yang; Yushui Geng, "A Survey of Anomaly Detection Methods in Networks", Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium.