To Reduce the False Alarm in Intrusion Detection System using self Organizing Map

Ritu Ranjani Singh
Technocrats Institute of technology
Bhopal, India
ritu_ranjani@yahoo.co.in

Prof.Neetesh Gupta
Technocrats Institute of technology
Bhopal, India
gupta.neetesh81@gmail.com

***Abstract*— Intrusion detection systems aim to identify attacks with a high detection rate and a low false alarm rate. Classification-based data mining models for intrusion detection are often ineffective in dealing with dynamic changes in intrusion patterns and characteristics. Consequently, unsupervised learning methods have been given a closer look for network intrusion detection. Traditional instance-based learning methods can only be used to detect known intrusions, since these methods classify instances based on what they have learned. They rarely detect new intrusions since these intrusion classes has not been able to detect new intrusions as well as known intrusions. In this paper, we propose a soft Computing technique such as Self organizing map for detecting the intrusion in network intrusion detection. Problems with k-mean clustering are hard cluster to class assignment, class dominance, and null class problems. The network traffic datasets provided by the NSL-KDD Data set in intrusion detection system which demonstrates the feasibility and promise of unsupervised learning methods for network intrusion detection.**

***Keywords*— Intrusion detection system, neural network, data mining, False alarm, Self organizing map**

I. Introduction

Intrusion is the sequence of the set of related activity which perform unauthorized access to the useful information and unauthorized file modification which causes harmful activity. Intrusion detection system deal with supervising the incidents happening in computer system or network environments and examining them for signs of possible events, which are infringement or imminent threats to computer security, or standard security practices Intrusion detection systems (IDS) have emerged to detect actions which endanger the integrity, confidentiality or availability of a resource as an effort to provide a solution to existing security issues. This technology is relatively new, however, since its beginnings, an enormous number of proposals have been put forward to sort this situation out in the most efficient and cost effective of manners [4].There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection, refer to Fig. 1. Anomaly detection identifies activities that vary from established patterns for users, or groups of users. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities. The second general approach to intrusion detection is misuse detection. This approach involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection approach frequently utilize a rule-based approach. When applied to misuse detection, the rules become scenarios for network attacks. The intrusion detection mechanism identifies a potential attack if a user's activities are found to be consistent with the established rules. The use of comprehensive rules is critical in the application of expert systems for intrusion detection [1]. A number of approaches based on computing have been proposed for detecting network intrusions. The guiding principle of soft computing is exploiting the tolerance of imprecision, uncertainty, partial robustness and low solution cost. Soft computing includes many theories such as Fuzzy Logic (FL), Artificial Neural Networks (ANNs), Probabilistic Reasoning (PR), and Genetic Algorithms (GAs). When used for intrusion detection, soft computing is a general term for describing a set of optimization and processing techniques that are tolerant of imprecision and uncertainty. Soft computing is often used in conjunction with rule-based expert systems where the knowledge is usually in the form of if-then rules. Despite different soft computing based approaches having been proposed in recent years, the possibilities of using the techniques for intrusion detection are still underutilized [7]. Some early research on IDSs explored neural networks for intrusion detection. These can be used only after training on normal or attack behaviors, or combination of the two. Most supervised neural net architectures require retraining to improve analysis on varying input data, unsupervised nets, which offer greater adaptability, can improve their analysis capability dynamically [8]. The majority of currently existing IDS face a number of challenges such as low detection rates and high false alarm rates, which falsely classifies a normal connection as an attack and therefore obstructs legitimate user access to the Network resources. These problems are due to the sophistication of the attacks and their intended similarities to normal behavior. More intelligence is brought into IDS by means of Machine Learning (ML). Theoretically, it is possible for a ML algorithm to achieve the best performance, i.e. it can minimize the false alarm rate and maximize the detection accuracy. However, this normally requires infinite training sample sizes (theoretically). In practice, this condition is impossible due to limited computational power and real-time response requirement of IDS. IDS must be active at any time and they cannot allow much delay because this would cause a bottleneck to the whole network [9].

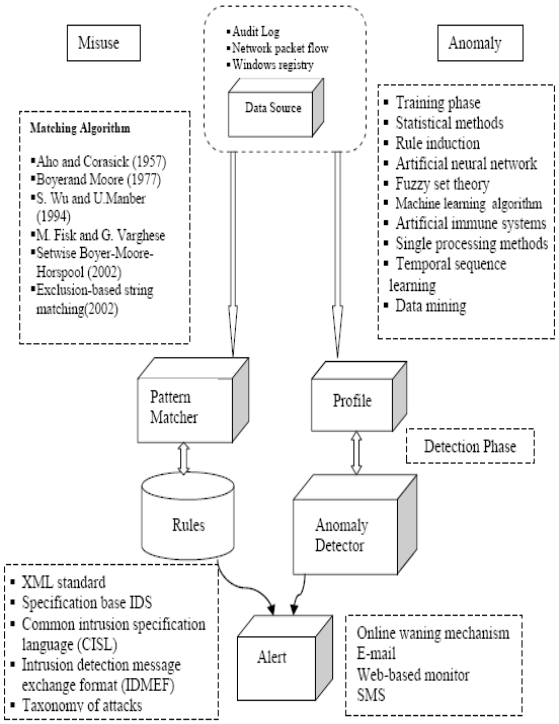


Fig. 1. The Flow Chart of Misuse Detection and Anomaly Detection Application [10].

To overcome low detection rate and high false alarm problems in currently existing IDS, we propose a hierarchical off line Anomaly intrusion detection system using Distributed Time-Delay Artificial Neural Network to enhance the performance of intrusion detection for rare and complicated attacks. In this paper, we introduce anomaly intrusion detection system, this can detect network-based attacks using dynamic neural nets, and has facilities for training, testing, and tuning of dynamic nets for intrusion detection purpose.

II. Related work on IDS

The goal for using ANNs for intrusion detection is to be able to generalize from incomplete data and to be able to classify data as being normal or intrusive. An ANN consists of a collection of processing elements that are highly interconnected. Given a set of inputs and a set of desired outputs, the transformation from input to output is determined by the weights associated with the inter-connections among processing elements. By modifying these interconnections, the network is able to adapt to desired outputs. The ability of high tolerance for learning-by-example makes neural networks flexible and powerful in IDS.IDS are split into two categories: misuse detection systems and anomaly detection systems(Anderson, 1980; Endorf et al., 2004). Misuse detection is used to identify intrusions that match known attack scenarios. However, anomaly detection is an attempt to search for malicious behavior that deviates from established normal patterns. In this paper our interesting is in anomaly detection. In order to detect the intrusion, various approaches have been developed and proposed over the last decade (Depren, Topallar, Anarim, & Ciliz, 2005; Patcha & Park, 2007). In the early stage, rule-based expert systems and statistical approaches are two typical ways to detect intrusion. A rule-based expert IDS can detect some well-known intrusions with high detection rate, but it is difficult to detect new intrusions, and its signature database needs to be updated manually and frequently (Lindqvist & Porras, 1999). Statistical-based IDS, employs various statistical methods including Principal component analysis (Shyu, Chen, Sarinnapakorn, & Chang, 2003), cluster and multivariate analysis (Taylor & Alves- Foss, 2001), Bayesian analysis (Barbard, Wu, & Jajodia, 2001), and frequency and simple significance tests (Qin & Hwang, 2004). But this type of IDS needs to collect enough data to build a complicated mathematical model, which is impractical in the case of complicated network traffic (Gordeev, 2000).To solve the limitations of above methods, a number of data mining techniques have been introduced (Dokas et al., 2002; Wu & Yen, 2009). Among these techniques, ANN is one of the most used techniques and has been successfully applied to intrusion detection (Horeis, 2003; Joo et al., 2003; Kevin, Rhonda, & Jonathan, 1990; Tan, 1995). According to different types of ANN, theseTechniques can be classified into the following three categories: supervised ANN-based intrusion detection, unsupervised ANN based intrusion detection, and hybrid ANN-based intrusion detection. Supervised ANN applied to IDS mainly includes multi-layer feed-forward (MLFF) neural networks and recurrent neural networks (Mukkamala, Janoski, & Sung, 2002). Ryan et al. (1998) and Tan (1995) used MLFF neural networks for anomaly detection based on user behaviors. But in practice the number of training set is very large and the distribution of training set is imbalanced, the MLFF neural networks is easy to reach the local minimum and thus stability is lower. Especially, for low-frequent attacks, the detection precision is very low. Some researchers have compared the effectiveness of supervised ANN with other methods such as support vector machine (SVM) and multivariate adaptive regression splines (MARS) (Mukkamala, Sung, Abraham, & Ramos, 2004; Mukkamala et al., 2002). Supervised ANN had been shown to have lower detection performance than SVM and MARS. The second category uses unsupervised ANN to classify input data and separate normal behaviors from abnormal or intrusive ones (Endorf et al., 2004). Using unsupervised ANN in intrusion detection has many advantages. The main advantage is that unsupervised ANN can improve their analysis of new data without retraining. The third category is hybrid ANN which combines supervised ANN and unsupervised ANN, or combine ANN with other data mining techniques to detect intrusion (Han & Cho, 2005; Jirapummin, Wattanapongsakorn, & Kanthamanon , 2002). The motivation for using the hybrid ANN is to overcome the limitations of individual ANN. Jirapummin et al. (2002) proposed employing a hybrid ANN for both visualizing intrusions using Kohenen's SOM and classifying intrusions using a resilient propagation neural networks. Horeis (2003) used a combination of SOM and radial basis function (RBF) networks. The system offers generally better results than IDS based on RBF networks alone. Han and Cho (2005) proposed

an intrusion detection technique based on evolutionary neural networks in order to determine the structure and weights of the call sequences. Chen, Abraham, and Yang (2007) proposed hybrid flexible neural-tree-based IDS based on flexible neural tree, evolutionary algorithm and particle swarm optimization (PSO). Empirical results indicated that the proposed method is efficient. For ANN based intrusion detection, hybrid ANN has been the trend (Chen et al., 2007). But different ways to construct hybrid ANN will highly influence the performance of intrusion detection. Different hybrid ANN models should be properly constructed in order to serve different aims. Following this stream, we propose Neural Network based algorithm called Self organizing map to solve the two drawbacks of current K-Mean algorithm

III. Intrusion Detection System

An Intrusion Detection System (IDS) constantly monitors actions in a certain environment and decides whether they are part of a possible hostile attack or a legitimate use of the environment. The environment may be a computer, several computers connected in a network or the network itself. The IDS analyzes various kinds of information about actions emanating from the environment and evaluates the probability that they are symptoms of intrusions. Such information includes, for example, configuration information about the current state of the system, audit information describing the events that occur in the system (e.g., event log in Windows XP), or network traffic. Several measures for evaluating IDS have been suggested (Debar *et al.* 1999; Richards 1999; Spafford and Zamboni 2000; Balasubramaniyan *et al.* 1998). These measures Include accuracy, completeness, performance, efficiency, fault tolerance, timeliness, and adaptively. The more widely used measures are The True Positive (TP) rate, that is, the percentage of intrusive actions (e.g., error related pages) detected by the system, False Positive (FP) rate which is the percentage of normal actions (e.g., pages viewed by normal users) the system incorrectly identifies as intrusive, and Accuracy which is the percentage of alarms found to represent abnormal behavior out of the total number of alarms. In the current research TP, FP and Accuracy measures were adopted to evaluate the performance of the new methodology. There are IDS that simply monitor and alert and there are IDS that perform an action or actions in response to a detected threat. We'll cover each of these briefly. Intrusion detection system various type of given:

A. *Nids*

Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. Ideally you would scan all inbound and outbound traffic; however doing so might create a bottleneck that would impair the overall speed of the network.

B. *Hids*

Host Intrusion Detection Systems are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity is detected

C. *Signature Based*

A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

D. *Anomaly Based*

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network- what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other- and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline.

IV. Traditional Technique used in IDS

A. *Clustering Techniques*

Clustering analysis [5] is the process of partitioning data objects (records, documents, etc.) into meaningful groups or clusters so that objects within a cluster have similar characteristics but are dissimilar to objects in other clusters. Clustering can be viewed as unsupervised classification of unlabelled patterns (observations, data items or feature vectors), since no pre-defined category labels are associated with the objects in the training set. Clustering results in a compact representation of large data sets (e.g., collections of visited Web pages) by a small number of cluster centroids. Applications of clustering include data mining, document retrieval, image segmentation, and pattern classification (Jain *et al.* 1999). Thus, clustering of Web documents viewed by Internet users can reveal collections of documents belonging to the same topic. As shown by Sequeira and Zaki (2002), clustering can also be used for anomaly detection: normality of a new object can be evaluated by its distance from the most similar cluster under the assumption that all clusters are based on 'normal' data only. A good clustering method will produce high quality clusters in which similarity is high known as intra-classes and inter-classes where similarity is low. The quality of clustering depends upon both the similarly measure used by the method and it, s implementation and it is also measured by the it, s ability to discover hidden patterns.The concept of clustering algorithms is to build a finite number of clusters, each one with its own center, according to a given data set, where each cluster represents a group of similar objects. Each cluster encapsulates a set of data and here the similarities of the surrounded data are their distance to the cluster center. Generally speaking, clustering techniques can

be divided into two categories pair wise clustering and central clustering. The former also called similarity-based clustering, groups similar data instances together based on a data-pair wise proximity measure. Examples of this category include graph partitioning-type methods. The latter, also called centroid-based or model-based clustering, represents each cluster by a model, i.e., its centroid".Central clustering algorithms [3] are often more efficient than similarity-based clustering algorithms. We choose centroid-based clustering over similarity-based clustering. We could not efficiently get a desired number of clusters, e.g., 100 as set by users. Similarity-based algorithms usually have a complexity of at least $O(N2)$ (for computing the data-pair wise proximity measures), where N is the number of data.

A. Hard Partitioning

These kind of methods are based on classical set theory and defines the presence or absence of a data point in a partition subset on strict logic, that is the object either belong to a subset or not. So, such kind of methods divides a dataset strictly into disjoint subsets. Conventional clustering [6] algorithms find a "hard partition" of a given data set based on certain criteria that evaluate the goodness of a partition. By hard partition we mean that each datum belongs to exactly one cluster of the partition. More formally, we can define the concept of "hard partition" as follows:

B. Soft Partitioning

A soft clustering algorithm partitions a given data set not an input space. Theoretically speaking, a soft partition not necessarily a fuzzy partition, since the input space can be larger than the dataset. In practice however most soft clustering Algorithms do generate a soft partition that also forms the fuzzy partition. A type of soft clustering of special interest is one that ensures the membership degree of a point x in all clusters adding up to one, i.e.

$$\sum_j \mu_{cj}(x_i) = 1, \ \forall \mathbf{x}_i \in \mathbf{X}$$

A soft partition that satisfies this additional condition is called a constrained soft partition. The fuzzy c-means algorithm produces a constrained soft partition. The fuzzy c-means algorithm is best known algorithm that produces constrained soft partition. The biggest drawback of a hard partitioning is the concept that it either includes a data point in a partition or strictly excludes it; there is no other chance for the data elements to be part of more than one partition at the same time. However, in natural clusters it is always the case that some of the data element partially belong to one set and partially to one or more other sets. In order to overcome this limitation, the notion of fuzzy partitioning was introduced [5].

K-means algorithm:

The K-means clustering is a classical clustering algorithm. After an initial random assignment of example to K clusters, the centres of clusters are computed and the examples are assigned to the clusters with the closest centres. The process is repeated until the cluster centres do not significantly change. Once the cluster assignment is fixed, the mean distance of an example to cluster centres is used as the score. Using the K-means clustering algorithm, different clusters were specified and generated for each output class

Input: The number of clusters K and a dataset for intrusion detection

Output: A set of K-clusters that minimizes the squared –error criterion.

Algorithm:

1. Initialize K clusters (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. Add element to cluster with min (Distance (xi, yj)).
4. Calculate the means of the clusters.
5. Change cluster centroids to means obtain Using Step 3.

The Main Disadvantage of K-Mean algorithm is that algorithm may take a large number of iterations through dense data sets before it can converge to produce the optimal set of centroids. This can be inefficient on large data sets due to its unbounded convergence of cluster centroids.

v. Proposed Approach

A. Soft computing techniques:-

Soft computing refers to a collection of computational techniques. Machine learning and some engineering disciplines, which study, model, and analyze very complex phenomena: those for which more conventional methods have not yielded low cost, analytic, and complete solutions. The two major problems –solving technologies include hard computing and soft computing .Hard Computing deals with precise models where accurate solutions are achieved quickly on the other side, soft computing deals with approximate Models and give solution to complex problems. The soft computing is a relatively new concept, and was introduced by Professor Lotfi Zadeh with the objective of exploiting the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, no solution cost and better rapport with reality.

B. Self-Organizing Map (SOM)

The Self-Organizing Map is a neural network model for analyzing and visualizing high dimensional data. It belongs to the category of competitive learning network. The SOM defines a mapping from high dimensional input data space onto a regular two-dimensional array of in designed architecture is input vector with six input values and output is realized to 2 dimension spaces. Every neuron i of the map is associated with an n-dimensional reference vector $[m_1, m_2..m_n]$ where n denotes the dimension of the input vectors. The reference vectors together form a codebook. The neurons of

the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or the structure, of the map. Adjacent neurons belong to the neighborhood Ni of the neuron i. In the SOM algorithm, the topology and the number of neurons remain fixed from the beginning. The number of neurons determines the granularity of the mapping, which has an effect on the accuracy and generalization of the SOM. During the training phase, the SOM forms elastic net that is formed by input data. The algorithm controls the net so that it strives to approximate the density of the data. The reference vectors in the codebook drift to the areas where the density of the input data is high.

C. The Learning Algorithm of the SOM

There are some basic steps involved in the application of the SOM algorithm. Firstly the weights of the network should be initialized. Assigning them small values picked from a random number generator can do this; in doing so, no prior order is imposed on the feature of map. The only restriction is that the weight vector, wj(0) should be different for j = 1, 2, . . . , n, where n is the number of neurons in the lattice. An alternative way of initializing the weight vector is to select from the available set of input vectors in a random manner. The key point is to keep the magnitude of the weights small, because the initial weights already give good approximation of the SOM weights. Next step is the similarity matching. With the use of the Euclidean minimum-distance criterion, the distance from the training data set to all weight vectors are computed and based on these computations the BMU is found. The Euclidean formula is given by

$$I(x)=\operatorname{argmin}||x(n)-w_j||,\ j=1,2\ldots n$$

Where I(x) identifies the best matching neuron to the input vector x. In words this formula finds the weight vector most similar to the input vector, x. This process sums up the essence of the competition among the neurons. In the sense of network topology there is a mapping process involved with this competition; A continuous input space of activation patterns is mapped onto a discrete output space of neurons by a process of competition among the neurons of the network [6].After having found the winning neuron the next step of the learning process is the updating. The weight vector of the winning neuron and the neurons close to it in the SOM lattice are adjusted towards the input vector. The update formula for the neuron j at time (i.e., number of iteration) n with weight vector w_j (n) is

$$w_j(n+1)=w_j(n)+\eta(n)h_{j,i(x)}(n)(x-w_j(n))$$

Where η(n) is the learning-rate parameter and hj, i(x) (n) is the time-varying neighborhood function centered around the winning neuron I(x). A typical choice of hj,i(x) is the Gaussian function [7] which is given by the formula

$$h_{i,i(x)}(n)=\exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right)$$

Where η(n) is a width function and dj,i represents the distance between the winning neuron i and its neighbor neuron j. The whole process is repeated for each input vector over and over for a number of cycles until no noticeable changes in the feature map are observed or a certain number of approaches are reached. Where η(n) is the learning-rate parameter [8] and hj, i(x) (n) is the time-varying neighborhood function centered around the winning neuron i(x).

D. Evaluation Measures

To evaluate the system performance the following measures (based on Sequeira and Zaki 2002) were used.

True Positive Rate (TP):also known as Detection Rate or Completeness): the percentage of terrorist pages receiving a rating above the threshold in the experiments, terrorist pages will be obtained from the users simulating terrorists.

False Positive Rate **(FP):** the percentage of regular Internet access pages that the system incorrectly determined as related to terrorist activities, i.e., the percentage of non-terrorist pages receiving a rating above threshold and suspected falsely as terrorists.

Accuracy**:** percentage of alarms related to terrorist behavior out of the total number of alarms. Since no benchmark data on content based intrusion detection is currently available, the results are compared to the best numbers achieved with ADMIT which is a command level method using the Means clustering algorithm to detect intruders (Sequeira and Zaki 2002).

VI. Experiment

A. Data Set Description

Because the goal of this work is to study and enhance the learning capabilities of the neural network techniques for intrusions detection, the Self organizing map method is compared to a clustering based k-mean algorithm that use the full set of samples sampled from the –NSL-KDD train dataset and witch contain 5000 sample. The original data set contain 5 million records which specify various attacks in which 1% sample consisting of about 5000 records was used in our experiment.

B. Data Preprocessing

In K-DD-98 data set, each records representing a connection between two networks host according to some well defined network protocol. Each connection is represented by 41 features, which include the basic features of individual of TCP Connections, the content features, No. Of byte, Transferred byte etc. the features in column 2 in –NSL-KDD Data set are transmitted byte, flag. We are interested in anomaly detection via unsupervised Learning algorithm. Hence all the records labeled as attack were considered as intrusion, while remaining was considered as normal. The labels are not used during clustering process but are used for evaluating the detection performance of the algorithm. The

clustering algorithm used in our comparative study do not handle categorical data the categorical feature such as flag in the data set are converted using 1-to-N Encoding Scheme.

C. Empirical Setting

The k-means and Self-organizing map algorithms are written in visual basic and compiled into mix files. SOM algorithms are relatively efficient due to vectored programming and active optimization. All experiments are run on a PC with a 3.06GHz Pentium-4 CPU with 1GB DRAM and running Windows XP.For the online Self organizing map algorithm, the learning rate follows $m = 0.5$ In order to study the effect of the total number of clusters on the intrusion detection results, we performed empirical studies with 100 and 200 total numbers of clusters.For clustering quality, we use the Computation time and energy. The purity of a cluster is defined as the percentage of the most dominated instance category in the cluster, and average purity is the mean over all clusters. Its value can range from 0 to 1, with higher values representing better average purity.The run time of each algorithm is also recorded and compared. Each experiment is run ten times for evaluating intrusion detection results, we report false positive rate (fpr), and attack Detection rate .The false positive rate is the percentage of normal instances that are labeled as attacks. The attack detection rate represents the percentage of all attack instances that are detected, i.e., labeled as attacks. We also report in Graph, by varying the parameter *used* in the detection method, to show the tradeoff between the false positive rate and the detection rate.

D. Experimental Results

Now, we compare the aforementioned clustering algorithms on the whole data set (with 5000 data set The Computation time results for the clustering algorithms with 100 clusters are shown in Table 1 respectively.In this experiment, we investigate computation time of SOM and K-Mean algorithm. In the training phase, the SOM were used to cluster the training data. After training, each cluster was labeled according to the majority type of data in this cluster. For instance, if more than 50% of the connections in cluster were intrusions, the cluster and its centroid weight vector would be labeled as intrusion.Self-organizing map perform significantly better ($p < 5\%$) than the others in terms of computation time with much less run time. Comparing the results for 100 clusters is shown in table 5.1. Self-organizing map algorithms perform significantly better ($p < 5\%$) than the others in terms of computation time. Comparing the results for 100 clusters, we observe that the k-means take more execution time than self-organizing map

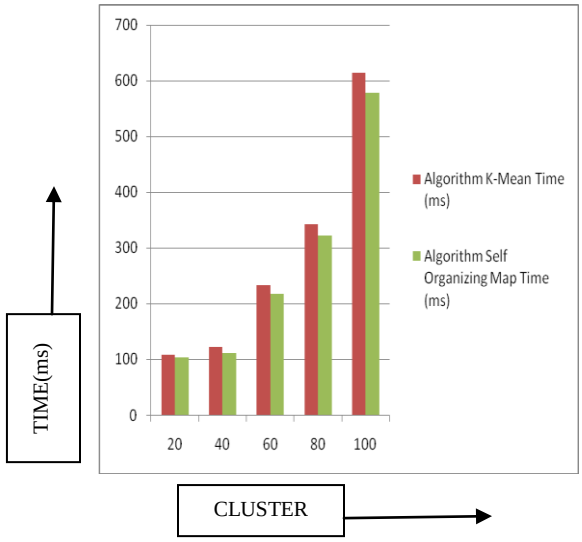
TABLE 6.4.1: SUMMARY OF CLUSTERING WITH 100 CLUSTER

Cluster	Algorithm	
	K-Mean	***Self Organizing Map***
	Time (ms)	Time (ms)
20	109	104
40	123	112
60	233	217
80	342	322
100	614	578

Since our aim is to detect network intrusion using clustering algorithms [10], we now analyze the unsupervised intrusion detection accuracies. We sort clusters according to their possibility of being normal in decreasing order and arrange data instances in a cluster in the same way. The distance to the centroid of the largest cluster measures the possibility of being normal. Graph can be constructed by dividing the sorted data instances into normal and intrusive categories at a series of cutting points. Now we find the false positive and detection rate of K-Mean and self-organizing map

TABLE 6.4.2: SUMMARY OF CLUSTERING RESULT WITH 100 CLUSTER

Algorithm	False positive	Detection rate
K-Mean algorithm	1%	60%
Self organizing map	0.5%	64%



. Fig (6.4): Graph for detection rate on 100 clusters

Fig. (6.4) Show the *Graph* for detection rate of the k-mean algorithms, and self-organizing map with 100 clusters respectively. The graph for the k-means and SOM algorithms are omitted for comprehensibility and better visualization, particularly because they are visibly worse. It can be seen that for 100 clusters, the SOM algorithm high detection rate the SOM algorithm clusters performs extremely well at very low

false positive rates, it can detect more than 64% attacks with a false positive rate of almost 1. Overall, the k-mean algorithm better ones and stable across different Number of clusters. 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 1% of the total population. We run each experiment 5 times and report the computation time, false positive rate, and attack detection rate. In Tables 6.4.1 and 6.4.2, self-organizing map better detection rate than k-mean algorithm and it take less computation time than k-mean algorithm. It performs the better when learning rate is less then 5%.

VII. Conclusion

In this work, we study the possible use of the neural networks learning capabilities in Intrusion Detection System. an innovative, knowledgebase Methodology for terrorist activity detection on the Web is presented. Neural Network methodology can be useful for detecting terrorists and their supporters using a legitimate ways of Internet access to view terror related content at a series of evasive web sites. The detection methodology res and Tables presented here can be applied to detecting to unseen attack.

References

[1] Abirami Muralidharan, J.Patrick Rousche, " Decoding of auditory cortex signals with a LAMSTAR neural network ", Neurological Research, Volume 27, pp. 4-10, January 2005

[2] Srilatha Chebrolu et.al, "Feature deduction and ensemble design of intrusion detection systems", Elsevier Journal of Computers & Security" Vol. 24/4, pp. 295-307, 2005

[3] Srilatha Chebrolu et.al, "Feature deduction and ensemble design of intrusion detection systems", Elsevier Journal of Computers & Security" Vol. 24/4, pp. 295-307, 2005

[4] H. Shah, J. Undercoffer, and A. Joshi, "Fuzzy Clustering for Intrusion Detection", *Proc. 12th IEEE Int'l Conf. Fuzzy Systems (FUZZ-IEEE '03)*, **2**, pp. 1274 – 1278, 2007.

[5] Leonid Portnoy, "Intrusion Detection with Unlabeled Data using Clustering", Undergraduate Thesis, Columbia University, New York, NY, Dec. 2007

[6]Wenke Lee, Sal Stolfo, and Kui Mok, "Adaptive Intrusion Detection: A Data Mining Approach", Artificial Intelligence *Review*, Kluwer Academic Publishers, 14(6): pp.533-567, December 2008.

[8] Michael Sobirey's Intrusion Detection Systems page, http://www.rnks.informatik.tucot.

[9]"NIST Special Publication on Intrusion Detection Systems", SP 800-31 Computer Security Resource Center (CSRC), National Institute of Standards and Technology (NIST), Nov. 2008, p.15.

[10]P.Lichodzijewski, A. n. Zincir-Heywood and M. I. Heywood, "Host-based intrusion detection using Neural Gas," Proceedings of the 2002 IEEE World Congress on Computational Intelligence, 2002 (in press).