An Efficient Approach to enhance the clustering and classification ensembles technique based on RBF and SOM Network

Kirit Singh, Prof. Sitendra Tamrakar

Abstract— In this paper we proposed a novel method for mixed data classification based on clustering and classification ensemble. Ensemble learning is a commonly used tool for building prediction models from data classification, due to its intrinsic merits of handling large volumes data. Despite of its extraordinary successes in stream data mining, existing ensemble models, in stream data environments, mainly fall into the ensemble classifiers category, without realizing that building classifiers requires labor intensive labeling process, and it is often the case that we may have a small number of labeled samples to train a few classifiers, but a large number of unlabeled samples are available to build clusters from mixed data. Ensemble clustering-classification aims to combine multiple clusters together for prediction. For a given test set, each cluster will derive a label vector. Noticing that some label vectors may conflict with each other, most state-of-theart ensemble clusters models employ a equiledian distance metric to minimize the discrepancy between each pair of label vectors. Although such a label vector based consensus method performs well on mixed dataset. Our novel approached divide into three sections first on ECC method second one is ECC with SOM network and finally ECC –RBF.

Index Terms— Clustering, ECC, SOM, RBF

1 Introduction

An intrusion can be defined [1, 2] as "any set of actions that attempts to compromise the integrity, confidentiality, or availability of a resource". User authentication (e.g., using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption) have all been used to protect computer systems. As systems become more complex, there are always exploitable weaknesses due to design and programming errors, or through the use of various "so-cially engineered" penetration techniques. For example, ex-ploitable "buffer overflow" still exists in some recent system software because of programming errors. Elements central to intrusion detection are resources to be protected in a target system, i.e., user accounts, file systems, system kernels, etc.; models that characterize the "normal" or "legitimate" behavior of these resources; techniques that compare the actual system activities with the established models identifying those that are "abnormal" or "intrusive". In pursuit of a secure system, different measures of system behavior have been proposed, on the basis of an ad hoc presumption that normalcy and anomaly (or illegitimacy) will be accurately manifested in the chosen set of system features.

Intrusion Detection attempts to detect computer attacks by examining various data records observed through processes on the same network. These attacks are split into two categories, host-based attacks [3, 1, 4] and network-based attacks [5, 6, 7]. Host-based attacks target a machine and try to gain access to privileged services or resources on that machine. Host-based detection usually uses routines that obtain system call data from an audit-process which tracks all system calls made on behalf of each user. Network-based attacks make it difficult for legitimate users to access various network services by purposely occupying or sabotaging network resources and services. This can be done by sending large amounts of network traffic, exploiting well known faults in networking services, overloading network hosts, etc. Network-based attack detection uses network traffic data (i.e., tcpdump) to look at traffic addressed to the machines being monitored. Intrusion detection systems are split into two groups, anomaly detection systems and misuse detection systems. Anomaly detection is the attempt to identify malicious traffic based on deviations from established normal network traffic patterns [7, 8]. Misuse detection is the ability to identify intrusions based on a known pattern for the malicious activity [5, 9]. These known patterns are referred to as signatures. Anomaly detection is capable of catching new attacks. However, new legitimate behavior can also be falsely identified as an attack, resulting in a false positive.

Network-based versus host-based IDS

A network-based IDS (NIDS) monitors the network traffic of a particular network. A host based IDS (HIDS) monitors the operating system, applications, and the host specific network traffic. They reside, at least partially, on a host. But some IDS's are of a hybrid type and implement parts of both approaches. See Table 1.1 for advantages and disadvantages. This thesis is only about NIDS's. In the following chapter of this thesis the term IDS is used Instead of NIDS.

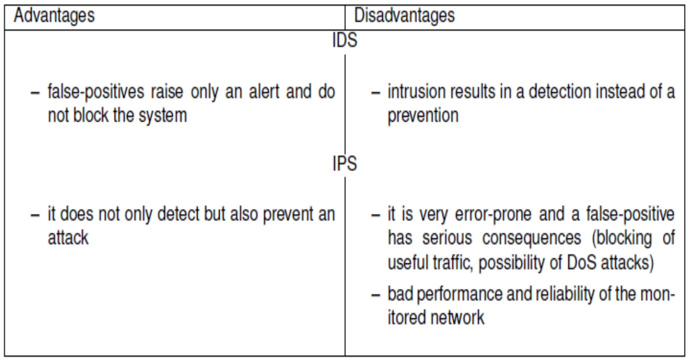
Advantage	*DisAdvantage*
Network Based	
• *No impact on the end system* • *Invisible Configuration* • *Detection of distributed attacks*	• *High requirement to scan every packet* • *Detection of attacks in the network attack* • *Can't used with encrypted message*
Host Based	
• *Monitors the actual reaction of host* • *Monitoring on all layer protocol* • *Encryption is no hindrance*	• *Installation on every single host* • *Performance requirement on every supervised host* • *No detection on distributed attack*

Intrusion prevention system (IPS) versus IDS

An IPS, also known as active IDS, investigates the traffic inline. This means that the packets Are analyzed continuously and the reaction to an attack is in real-time. The IPS blocks Traffic independently without human interaction. It aims not only at detecting, but also at preventing an attack. An IPS can be seen as an extended firewall, which does not inspect the packet headers alone (e.g. IP's and ports), but also other properties such as protocol Flow or the content of a packet. In contrast, a passive IDS does not act by itself but does only raise an alarm in case of a supposed attack. The handling of this alarm needs human interaction.

Signature-based versus behavior-based IDS

A signature-based or so-called misuse detection system searches for known malicious patterns in the payload. A pure signature-based IDS uses only single events for the analysis.A behavior-based IDS, also known as an anomaly detection system, analyses in the first instance the traffic data. The goal is to distinguish between normal and abnormal traffic examining the fundamental behaviour of a system. See Table 1.2 for advantages and disadvantages.In this thesis, we combine these two approaches. We implement a signature-based IDS using Snort and refine the escalated events using behavior-based correlation.



Advantages	Disadvantages
IDS	
– false-positives raise only an alert and do not block the system	– intrusion results in a detection instead of a prevention
IPS	
– it does not only detect but also prevent an attack	– it is very error-prone and a false-positive has serious consequences (blocking of useful traffic, possibility of DoS attacks) – bad performance and reliability of the monitored network

2 Literature Survey

ANALYSIS OF INTRUSION DETECTION SYSTEM USING VARIOUS NEURAL NETWORK CLASSIFIERS by S.Devaraju, Dr. S.Ramakrishnan (2011)

They proposed that the signature based intrusion is detected using neural network classifier like Feed Forward Neural Network (FFNN), Probabilistic Neural Network (PNN) And Radial Basis Neural Network (RBNN). The various techniques are applied in this problem in MATLAB application for improving the best performance applied to KDD Cup 1999 dataset. The performance of the full featured dataset and reduced dataset is analyzed [14]. The Figure 1 shown that the classification of proposed systems:

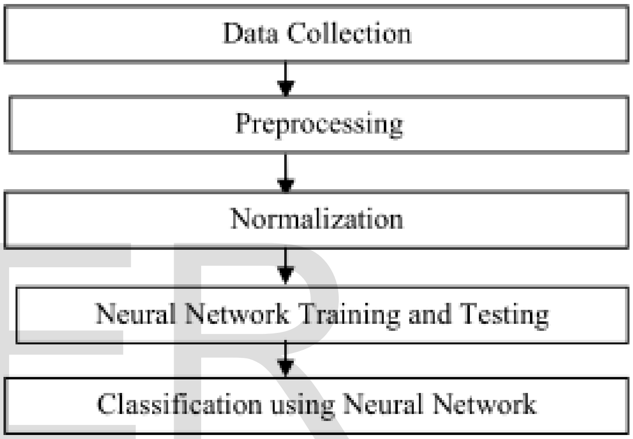

Data Collection
Preprocessing
Normalization
Neural Network Training and Testing
Classification using Neural Network

Figure 1

In the proposed research the three types of classifiers used are Feed Forward

Neural Network (FFNN), Probabilistic Neural Network (PNN) and Radial Basis Neural Network (RBNN). In this problem, the feature reduction techniques are used to a given KDD Cup 1999 dataset. The performance of the full featured KDD Cup 1999 dataset is compared with that of the reduced featured KDD Cup 1999 dataset. The MATLAB software is used to train and test the dataset and the efficiency is measured. Using the above said technique, it is proved that the reduced dataset is performing better than the full featured dataset.

Network Intrusion Detection Based on Improved Proximal SVM by Chengjie GU, 1Shunyi ZHANG, 2Xiaozhen XUE (2011)

Intrusion detection is one of the most essential factors for security infrastructures in network environments, and it is widely used in detecting, identifying and tracking the intruders. To solve the drawback of the SVM algorithm to meet the requirements of the network intrusion detection, we propose network intrusion detection based on improved proximal SVM. Experiment results illustrate the formulation of PSVM greatly simplifies the problem with considerably faster computational time than SVM for network intrusion. This method also can shorten the training time and improve detection performance by improved kernel function [15].

A Study of Intrusion Detection System Based on Data Mining by Chunyu Miao and Wei Chen (2010)

In this paper [16] , classifications of intrusion detection and methods of data mining applied on them were introduced. Then, intrusion detection system design and implementation of based on data mining were presented. Such a system used

APRIORI algorithm to analyse data association, which is the most influencing algorithm in mining Boolean association rules continuity item muster, with recurrence arithmetic based on idea of two period continuity item muster as core. Experiments showed that new type of attack can be detected effectively in the system, and knowledge base can be updated automatically, so the efficiency and accuracy of the intrusion detection were improved, and security of the network was enhanced.

A Novel Rule-based Intrusion Detection System by Lei Li, De-Zhang Yang, Fang-Cheng Shen (2010)

Network security is becoming an increasingly important issue, since the rapid development of the Internet. Network Intrusion Detection System (IDS), as the main security defending technique, is widely used against such malicious attacks. Data mining and machine learning technology has been extensively applied in network intrusion detection and prevention systems by discovering user behavior patterns from the network traffic data. Association rules and sequence rules are the main technique of data mining for intrusion detection [17]. Considering the classical Apriori algorithm with bottleneck of frequent itemsets mining, we propose a Length-Decreasing Support to detect intrusion based on data mining, which is an improved Apriori algorithm. Experiment results indicate that the proposed method is efficient.

An Enhanced Support Vector Machine Model for Intrusion Detection by JingTao Yao, Songlun Zhao, and Lisa Fan (2008)

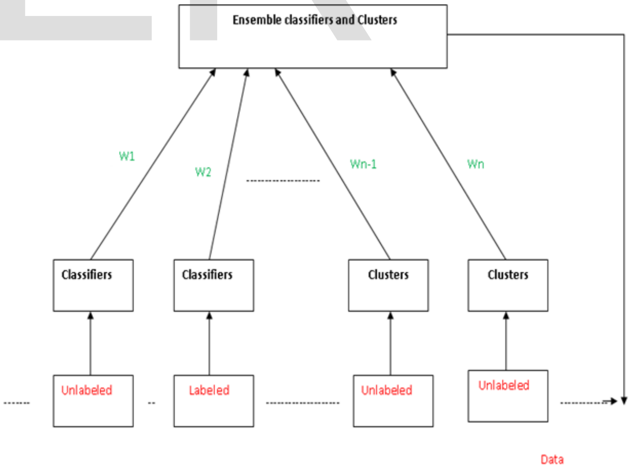
Design and implementation of intrusion detection systems remain an important research issue in order to maintain proper network security. Support Vector Machines (SVM) as a classical pattern recognition tool have been widely used for intrusion detection. However, conventional SVM methods do not concern di®erent characteristics of features in building an intrusion detection system. We propose an enhanced SVM model with a weighted kernel function based on features of the training data for intrusion detection. Rough set theory is adopted to perform a feature ranking and selection task of the new model. We evaluate the new model with the KDD dataset and the UNM dataset. It is suggested that the proposed model outperformed the conventional SVM in precision, computation time, and false negative rate [18].

A.M and A.S (2008) proposed a technique of combining K-Means clus tering and genetic algorithm to IDS. The training data has been clustered in t o 2-clusters before feeding the initial population hoping that data wil be divided into normal an d abnormal clusters. There were no declared experiment results but the author concluded that his approach detected known and unknown and the results were not good for some runs.

Liwei Ku an g (2007), proposed a Dependable Network Intrusion Detection System (DNIDS) based on the Combined Strangeness and Isolation measure K-Nearest Neighbor (CSIKNN) algorithm. The intrusion detection algorithm analyzes different characteristics of network data by employing two measures: strangeness and isolation. But in general the K-NN stil needs intensive computations. The Unsupervised Anomaly Detection Using an Optimized K-Nearest Neighbors Algorithm can work without the need for massive sets of pre-labeled training data. The author discussed the creation of such a system that uses a k-nearest neighbors algorithm to detect anomalies in network connections, as wel as the optimization necessary to make the algorithm feasible for a real-world system. The drawback of this approach is that the detection rates and false positive rates were not good as other approaches.

3 Proposed Methodology

ECC model



Algorithm:
In our proposed work we used standard deviation which calculates the error rate, which means how much our result is deviating from the exact result.

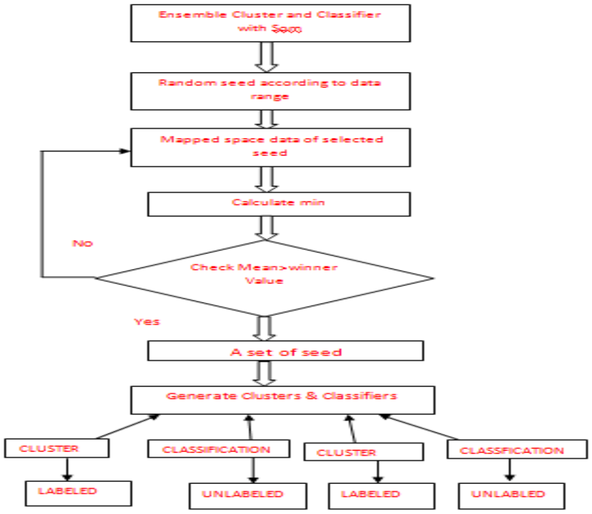
(1) Divide dataset into chunk D1, D2, D3, _____Dn+1.
(2) Generate discrete random number of seed for generating of cluster.
(3) Initialized distance weight factor.
(4) Calculate min of data chunk and standard deviation.
(5) Compare value at min with seed value.
(6) Then generate cluster.

(7) Set label of class C1, C2, C3.
(8) Assigned training at data.
(9) Generate classifier-Merge set of cluster & classifier with label.
(10) Calculate standard deviation (error).
(11) Ensemble class.
(12) Data classified.

Here this algorithm simply tells that initially there is a large data. Firstly it is divided in to the small size chunks. Then we us a random number generator which generate a random number is time. This random number works as a seed for each iteration. It means in each iteration we take a new random number. This iterative process generates a set of seed which is used to generate a clusters, i.e. each seed is works as a representative for each cluster.

ECC-SOM

In this phase of algorithm ensemble technique consider self organized network (SOM). Using SOM network control the generation of center point of cluster and iteration of cluster. The basic idea of a SOM is to map the data patterns onto a *n*-dimensional grid of neurons or units. That grid forms what is known as the output space, as opposed to the input space where the data patterns are. This mapping tries to preserve topological relations, i.e., patterns that are close in the input space will be mapped to units that are close in the output space, and vice-versa. So as to allow an easy visualization, the output space is usually 1 or 2 dimensional. Let X be the set of n training patterns x1, x2, xn W be a p×q grid of units wij where i and j are their coordinates on that grid α be the learning rate, assuming values in[0,1], initialized to a given initial learning rate r be the radius of the neighborhood function h(wij,wmn,r), initialized to a given initial radius.


Ensemble Cluster and Classifier with Seed
Random seed according to data range
Mapped space data of selected seed
Calculate min
No
Check Mean>winner Value
Yes
A set of seed
Generate Clusters & Classifiers
CLUSTER
CLASSIFICATION
CLUSTER
CLASSFICATION
LABELED
UNLABELED
LABELED
UNLABLED

Algorithm:

In this phase of algorithm SOM network apply on ECCA technique. In this process SOM control the iteration of selected data for clustering.

(1) Divide dataset into chunk D1, D2, D3, _____Dn+1.
(2) Generate discrete random number of seed for generating of cluster.
(3) Initialized distance weight factor.
(4) Map data into SOM space
(5) Calculate min of data chunk and standard deviation.
(6) Calculate Winner matrix
(7) Compare value at min with seed value.
(8) Repeat iteration
(9) Then generate cluster.
(10) Set label of class C1, C2, C3.
(11) Assigned training at data.
(12) Generate classifier-Merge set of cluster & classifier with label.
(13) Calculate standard deviation (error).
(14) Ensemble class.
(15) Data classified

ECC-RBF

In this section of method RBF network implied on clustering classification ensembles technique for better training purpose of data for improvement of classification rate of ensembles technique. A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin. If a function 'h' satisfies the property $h(\mathbf{x})=h(||\mathbf{x}||)$, then it is a radial function. Their characteristic feature is that their response decreases (or increases) monotonically with distance from a central point. The centre, the distance scale, and the precise shape of the radial function are parameters of the model, all fixed if it is linear [25]. A typical radial function is the Gaussian which, in the case of a scalar input, is

Its parameters are its centre *c* and its radius *r*. A Gaussian RBF monotonically decreases with distance from the centre. In contrast, a multiquadric RBF which, in the case of scalar input monotonically increases with distance from the centre. Gaussian-like RBFs are local (give a significant response only in a neighborhood near the centre) and are more commonly used than multiquadric-type RBFs which have a global Response. Radial functions are simply a class of functions. In principle, they could be employed in any sort of model (linear or nonlinear) and any sort of network (single-layer or multi-layer). RBF networks have traditionally been associated with radial functions in a single-layer network. In the Figure 4.4, the input layer carries the outputs of FLD function. The distance between these values and centre values are found and summed to form linear combination before the neurons of the hidden layer. These neurons are said to contain the radial basis function with exponential form. The outputs of the RBF activation function is further processed according to specific Requirements.

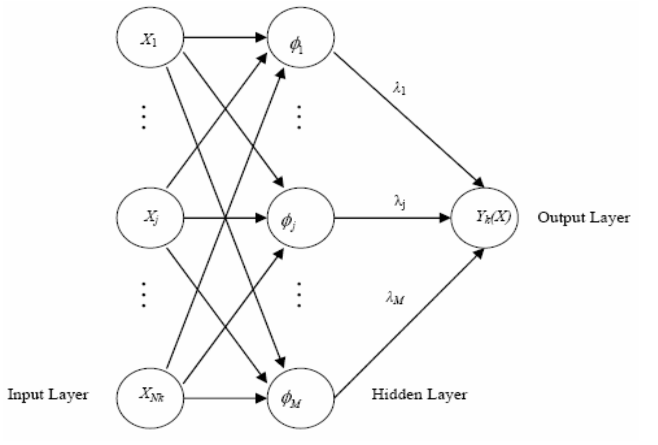


Figure 4.4. Structure of Radial Basis Function Neural Network.

In order to specify the middle layer of an RBF we have to decide the number of neurons of the layer and their kernel functions which are usually Gaussian functions. In this paper we use a Gaussian function as a kernel function. A Gaussian function is specified by its center and width. The simplest and most general method to decide the middle layer neurons is to create a neuron for each training pattern. However the method is usually not practical since in most applications there are a large number of training patterns and the dimension of the input space is fairly large. Therefore it is usual and practical to first cluster the training patterns to a reasonable number of groups by using a clustering algorithm such as K-means or SOFM and then to assign a neuron to each cluster. A simple way, though not always effective, is to choose a relatively small number of patterns randomly among the training patterns and create only that many neurons. A clustering algorithm is a kind of an unsupervised learning algorithm and is used when the class of each training pattern is not known. But an RBFN is a supervised learning network. And we know at least the class of each training pattern. So we'd better take advantage of the information of these class memberships when we cluster the training patterns. Namely we cluster the training patterns class by class instead of the entire patterns at the same time (Moody and Darken, 1989; Musavi et al., 1992). In this way we can reduce at least the total computation time required to cluster the entire training patterns since the number of patterns of each class is usually far less than that of the entire patterns. We use an one-pass clustering algorithm called APC-III (Hwang and Bang, 1994). APC-III is similar to RCE (Reilly et al., 1982) but different in that APC-III has a constant radius while RCE has a variable radius. First of all we decide the radius *R*0 of clusters. Therefore APC-III creates many clusters if the radius is small and few clusters if it is large. We set *R*0 to the mean minimum distance between the training patterns multiplied by a:

$$R_0 = \alpha \frac{1}{P} \sum_{i=1}^{P} \min_{i \neq j}(\|\mathbf{x}_i - \mathbf{x}_j\|)$$

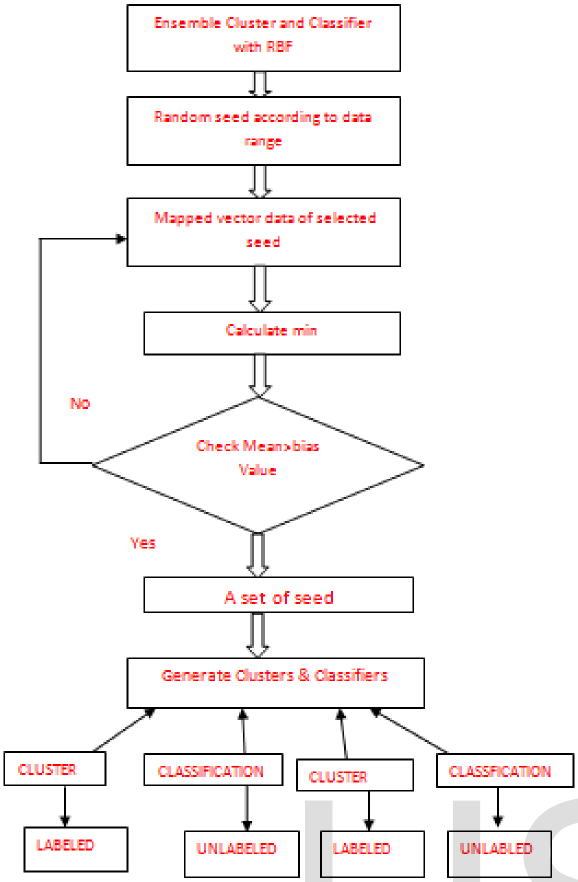
Where *P* is the number of the training patterns. If the number of the training patterns is too large, we may well use a subset of them to obtain an approximate *R*0 instead of the exact *R*0. This will speed up the calculation of *R*0. Next the following procedure is repeated to find clusters. If a given training pattern falls in the region of *R*0 of any existing cluster, we include it in the cluster by adjusting the center of the cluster as described in the algorithm below. By keeping only the number of the training patterns included in the cluster, we can readily calculate the new center of the cluster. If it falls in none of the existing clusters, we create a new cluster whose center is set to the given training pattern.

The outline of APC-III algorithm can be stated as follows:

Input: training patterns *X*== {x1; x2,…………,x*P}*
Output: centers of clusters
Variable
C: number of clusters
c*j* : center of the *j*-th cluster
nj : number of patterns in the *j*-th cluster
di j : distance between x*i* and the *j*-th cluster

```
begin
C =1; c1  x1;n1 :=1;
for i :=2 to P do /* for each pattern */
for j :=1 to C do /* for each cluster */
compute di j;
if di j _R0 then
/* include xi into the j-th cluster */
cj  (cjnj +xi)=(ni+1);
ni :=ni+1;
exit from the loop;
end if
end for
if xi is not included in any clusters then
/* create a new cluster */
C :=C+1;
cC  xi;
nC :=1;
end if
end for
end
```

APC-III is quite efficient to construct the middle layer of an RBF since we can finish clustering by going through the entire training patterns only once. This is not true with K-means and SOFM clustering algorithms. Furthermore APC-III tends to create an appropriate number of clusters since it determines the radius of a cluster based on the distribution of the training patterns. This fact makes APC-III to perform as good as the regular multi-pass clustering algorithms.


Ensemble Cluster and Classifier with RBF
Random seed according to data range
Mapped vector data of selected seed
Calculate min
Check Mean>bias Value
No
Yes
A set of seed
Generate Clusters & Classifiers
CLUSTER
CLASSIFICATION
CLUSTER
CLASSFICATION
LABELED
UNLABELED
LABELED
UNLABLED

4 Result Analysis

As seen from the output of performance on all following data sets. It can be made out that when ECC is combined with SOM and RBF method, the performance gets significantly improved.

Earlier application of isolated ECC on dataset has much greater Accuracy, than later by integrating both ECC and SOM and RBF Method SOM and RBF. Also there is a considerable enhancement in the true positive and true negative detection ratio and minimizes in false positive and false negative ratio .Thus this gives the direct improvised accuracy in the result. Basis the result of confusion matrix (true positive, true negative, false positive, false negative).We are showing the consequence for the following parameters i.e. - Accuracy, Precision, Recall for data sets.

Precision- Precision measures the proportion of predicted positives/negatives which are actually positive/negative.

Recall -It is the proportion of actual positives/negatives which are predicted positive/negative.

Accuracy-It is the proportion of the total number of prediction that were correct or it is the percentage of correctly classified instances.

Below we are showing how to calculate these parameters by the suitable formulas. And also, below we are showing the graph for that particular data set.

$$\text{Precision} = \frac{TP}{TP+FP}$$

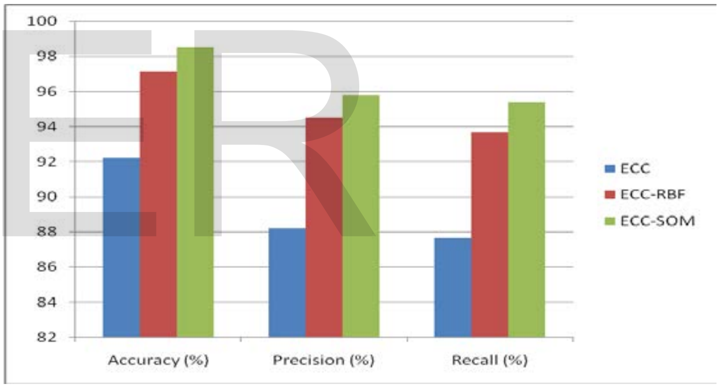
$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP}$$

$$\text{FPR}=\frac{FP}{FP+TN}\ ,\ \text{FNR}=\frac{FN}{FN+TP}$$

Chart of ECC and ECC-SOM consequence on given DataSet1 for Intrusion Detection System.

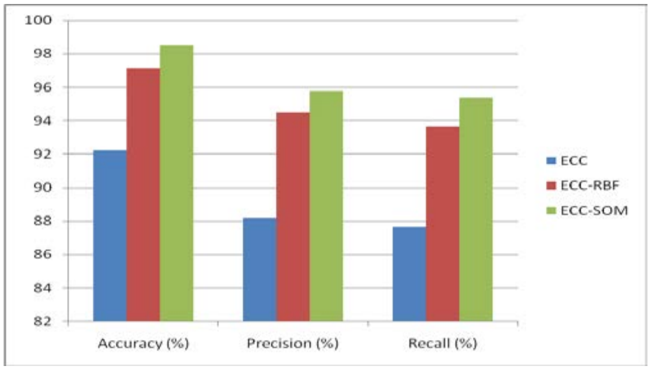
DATASET-1



Above graph represents the assessment result of DataSet1as ECC and ECC-SOM method as included parameters i.e. Accuracy, Precision, Recall.

Chart of ECC and ECC-SOM consequence on given DataSet2 for Intrusion Detection System.

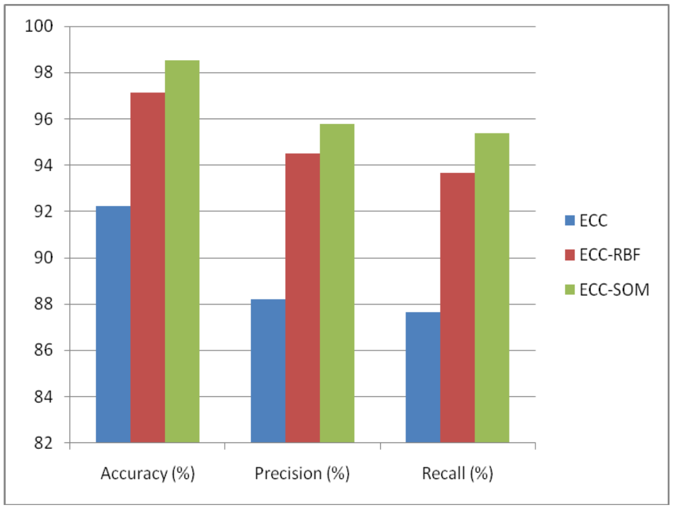
DATASET-2



Above graph represents the assessment result of Data-

Set2as ECC and ECC-SOM method as included parameters i.e. Accuracy, Precision, Recall.

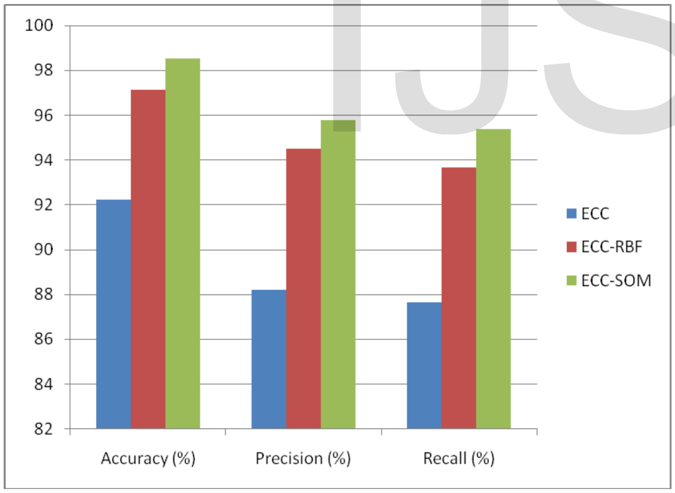
Chart of ECC and ECC-SOM consequence on given Data-Set3 for Intrusion Detection System.



Above graph represents the assessment result of Data-Set3 as ECC and ECC-SOM method as included parameters i.e. Accuracy, Precision, Recall.

Chart of ECC and ECC-SOM consequence on given Data-Set4 for Intrusion Detection System.

DATASET-4



Above graph represents the assessment result of Data-Set 4 as ECC and ECC-SOM method as included parameters i.e. Accuracy, Precision, Recall.

5 Cocnlusion

In this paper we proposed a novel method for mixed data classification based on clustering and classification ensemble. Ensemble learning is a commonly used tool for building prediction models from data classification, due to its intrinsic merits of handling large volumes data. Despite of its extraordinary successes in stream data mining, existing ensemble models, in stream data environments, mainly fall into the ensemble classifiers category, without realizing that building classifiers requires labor intensive labeling process, and it is often the case that we may have a small number of labeled samples to train a few classifiers, but a large number of unlabeled samples are available to build clusters from mixed data. Ensemble clustering-classification aims to combine multiple clusters together for prediction..

The KDD data cup set has been used for the evaluation of work proposed in this paper.

References

[1] S.Devaraju, Dr.S.Ramakrishnan: “ANALYSIS OF INTRUSION DETECTION SYSTEM USING VARIOUS NEURAL NETWORK CLASSIFIERS” pp 1033-1038, 2011.

[2] Network Intrusion Detection Based on Improved Proximal SVM by Chengjie GU, 1Shunyi ZHANG, 2Xiaozhen XUE (2011)

[3] A Study of Intrusion Detection System Based on Data Mining by Chunyu Miao and Wei Chen (2010)

[4] A Novel Rule-based Intrusion Detection System by Lei Li, De-Zhang Yang, Fang-Cheng Shen (2010)

[5] An Enhanced Support Vector Machine Model for Intrusion Detection by JingTao Yao, Songlun Zhao, and Lisa Fan (2008)

[6] Qu, X., S. Hariri, M. Yous if, “An Efficient Network Intrusion Detection Method Bas ed on Information Theory and Genetic Algorithm”, Proceedings of the 24th IEEE International Performance Computing and Communications

[7] Song, D., M.1. He ywo o d , A.N. Zincir-Heywood, 2005. Training Genetic Programming on Half a Million.

[8] Athanas ios Papoulis and S. Unnikrishna Pilla i, 2002. "Probability, Random Variables and stochas tic Processes ", Book. Book written by Athanasios Papoulis and S. Un n ikris h na Pillai, 2002. "Probability, Random Variables and s tochas tic Proces ses .

[9] Chi-Ho Ts ang, Sam Kwong, 2005. Hanli Wang1, Anomaly Intrusion Detection using Multi-Objective