SOME CLUSTERING ALGORITHMS TO ENHANCE THE PERFORMANCE OF THE NETWORK INTRUSION DETECTION SYSTEM

[1]Mrutyunjaya Panda, [2]Manas Ranjan Patra
[1] Department of E&TC Engineering, GIET, Gunupur, India
[2] Department of Computer Science, Berhampur University, India
E-mail: [1]panda_mrutyunjaya@yahoo.com, [2]mrpatra12@gmail.com

ABSTRACT

Most current intrusion detection systems are signature based ones or machine learning based methods. Despite the number of machine learning algorithms applied to KDD 99 cup, none of them have introduced a pre-model to reduce the huge information quantity present in the different KDD 99 datasets. Clustering is an important task in mining evolving data streams. Besides the limited memory and one-pass Constraints, the nature of evolving data streams implies the following requirements for stream clustering: no assumption on the number of clusters, discovery of clusters with arbitrary shape and ability to handle outliers. 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 some clustering algorithms such as K-Means and Fuzzy c-Means for network intrusion detection. The experimental results obtained by applying these algorithms to the KDD-99 data set demonstrate that they perform well in terms of both accuracy and computation time.

Key-Words: Intrusion Detection, K-Means, Fuzzy c-Means, MF plot, ROC

1. INTRODUCTION

An intrusion detection system (IDS) is a component of the information security framework. Its main goal is to differentiate between normal activities of the system and behaviour that can be classified as suspicious or intrusive [1]. The goal of intrusion detection is to build a system which would automatically scan network activity and detect such intrusion attacks. Once an attack is detected, the system administrator can be informed who can take appropriate action to deal with the intrusion.

IDS can be host-based (HIDS), network-based (NIDS) or a combination of both types (Hybrid Intrusion Detection System). HIDS usually observes logs or system –calls on a single host, while a NIDS typically monitors traffic flows and network packets on a network segment, and thus observes multiple hosts simultaneously. Generally, one deal with very large volumes of network data, and thus it is difficult and tiresome to classify them manually in order to detect a possible intrusion. One can obtain labelled data by simulating intrusions, but this will be limited only to the set of known attacks. Therefore, new types of attacks that may occur in future cannot be handled, if those were not part of the training data. Even with manual classification, we are still limited to identifying only the known (at classification time) types of attacks, thus restricting our detection system to identifying only those types.

To solve these difficulties, we need a technique for detecting intrusions when our training data is unlabeled, as well as for detecting new and un-known types of intrusions. A method that offers promise in this task is anomaly detection. Anomaly detection detects anomalies in the data (i.e. data instances in the data that deviate from normal or regular ones). It also allows us to detect new types of intrusions, because these new types will, by assumption, be deviations from the normal network usage.

It is very difficult, if not impossible, to detect malicious intent of someone who is authorized to use the network and who uses it in a seemingly legitimate way. For example, there is probably no highly reliable way to know whether someone who correctly logged into a system is the intended user of that system, or if the password was stolen.

Under these assumptions we built a system which created clusters from its input data, then automatically labelled clusters as containing either normal or anomalous data instances, and finally used these clusters to classify network data instances as either normal or anomalous. Both the training and testing was done using 10% KDDCup'99 data [2], which is a very popular and widely used intrusion attack dataset.

Most clustering techniques assume a well defined distinction between the clusters so that each pattern can only belong to one cluster at a time. This supposition can neglect the natural ability of objects existing in multiple clusters. For this reason and with the aid of fuzzy logic, fuzzy clustering can be employed to overcome the weakness. The membership of a pattern in a given cluster can vary between 0 and 1. In this model a data object belongs to the cluster where it has the highest membership value.

In this paper we aim to propose a fuzzy c-means clustering technique which is capable of clustering the most appropriate number of clusters based on objective function. This, as the name implies, draws the fuzzy boundary, thereby proving efficient when compared with that of its counterpart.

The rest of the paper is organised as follows. In section 2, we discuss Clustering methods; followed by Data Clustering algorithms in section 3. Section 4 describes about the experimental set-up and results obtained. Some discussion is made in section 5. Finally, section 6 provides some related works followed by conclusion in section 7.

2. CLUSTERING METHODS

Clustering may be found under different names in different contexts, such as unsupervised learning (in pattern recognition), numerical taxonomy (in biology, ecology), typology (in social sciences) and partition (in graph theory) [3].By definition, "cluster analysis is the art of finding groups in data", or from Wikipedia [4], "clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data into subsets (clusters), so that the data in each subset (ideally) share some common trait-often proximity according to some defined distance measure. Clustering is a challenging field of research as it can be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Alternatively, cluster analysis serves as a pre-processing step for other algorithms, such as classification which would then operate on detected clusters.

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. By clustering, one can identify dense and sparse regions and therefore, discover overall distribution patterns and interesting correlations among data attributes. Clustering does not rely on predefined classes and class labelled training examples. For this reason, clustering is a form of learning by observation.

In intrusion detection, an object is a single observation of audit data and/or network packets after the values from selected features have been extracted. Hence, values from selected features, and one observation, define one object (or vector). If we have values from n number of features, the vector (or object plot) fits into an n-dimensional coordinate system (Euclidean space R^n).

In order to derive the objective function and other relevant mathematics for fuzzy c-means and the remaining of its variations, it is better to see the same for the hard (crisp) partitioning technique, so that we may be able to understand the difference between the two approaches. (If we look into these issues all of them appears to be objective functional minimization problems. If the constraints are relaxed we get the possibilistic partition scheme. So, the clustering algorithm is nothing but a minimization problem which may be constrained or unconstrained.)

2.1. 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 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:

1) Let X be a data set of data, and x_i be an element of X. A partition p={$C_{1,}C_{2,}\ldots,C_j$} of X is "hard" if and only if

i) $\forall x_i \in X \quad \exists C_j \in P$ such that $x_i \in C_j$

ii) $\forall x_i \in X \quad x_i \in C_j \Rightarrow x_i \notin C_{k,}$

Where $k \neq j$, $C_j \in P$.

The first condition in the definition assures that the partition covers all data points in X; the second condition assures that all the clusters in partition are manually exclusive.

2) Let x be a data set of data and x_i be an element of X. A partition p={$C_{1,}C_{2,}\ldots,C_j$} of X is "soft" if and only if the following condition holds:

i) $\forall x_i \in X \quad \forall C_j \in P$ for $0 \le \mu_c(x_i) \le 1$;

ii) $\forall x_i \in X \quad \forall C_j \in P$ such that $\mu_{cj}(x_i)>0$.

2.2. 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.

$$\sum_j \mu_{cj}(x_i) = 1, \ \forall x_i \in X\text{------ (1)}$$

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 elements 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].

3. DATA CLUSTERING ALGORITHMS

The following are the algorithms used for clustering the datasets:

- K-means Algorithm
- Fuzzy c-means Algorithm.

3.1. K-means Clustering

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 [6].

K-means clustering is a well known Data Mining algorithm that has been used in an attempt to detect anomalous user behaviour, as well as unusual behaviour in network traffic. There are two problems that are inherent to K-means clustering algorithms. The first is determining the initial partition and the second is determining the optimal number of clusters [7]. The figure1 as shown below depicted the K-means algorithm.

As the algorithm iterates through the training data, each cluster's architecture is updated. In updating clusters, elements are removed from one cluster to another. The updating of clusters cause the values of the centroids to change. This change is a reflection of the current cluster elements. Once there are no changes to any cluster, the training of the K-Means algorithm is complete.

```
K-MEANS ALGORITHM:

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 ($x_i$, $y_j$)).
    4. Calculate the means of the clusters.
    5. Change cluster centroids to means
       obtained using Step 3.
```

Figure1. K-Means Clustering

At the end of the K-Means training, the K cluster centroids are created and the algorithm is ready for classifying traffic. For each element to be clustered, the cluster centroids with the minimal Euclidean distance from the element will be the cluster for which the element will be a member.

After training, the cluster centroids remains the same, like the SOM (Self organise Map) can be useful for anomaly detection tool that requires the input to remain static. The k-Means 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.

3.2. Fuzzy c-Means (FCM) Clustering

Fuzzy c-Means (FCM) algorithm, also known as fuzzy ISODATA, was introduced by Bezdek [8] as extension to Dunn's [9] algorithm to generate fuzzy sets for every observed feature. The fuzzy c-means clustering algorithm is based on the minimization of an objective function called c-means functional.

Fuzzy c-means algorithm is one of the well known relational clustering algorithms. It partitions the sample data for each explanatory (input) variable into a number of clusters. These clusters have "fuzzy" boundaries, in the sense that each data value belongs to each cluster to some degree or other. Membership is not certain, or "crisp". Having decided upon the number of such clusters to be used, some procedure is then needed to location their centres (or more generally, mid-points) and to determine the associated membership functions and the degree of membership for the data points.

Fuzzy clustering methods allow for uncertainty in the cluster assignments. FCM is an iterative algorithm to find cluster centres (centroids) that minimize a dissimilarity function. Rather that partitioning the data into a collection of distinct sets by fuzzy partitioning, the membership matrix (U) is randomly initialized according to equation 2.

$$\sum_{i=1}^{c} u_{ij} = 1,\ \forall_{j}=1,2,\ldots,n. \quad (2)$$

The dissimilarity function (or more generally the objective function), which is used in FCM in given equation 3.

$$J(U,c_1,c_2,\ldots,c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} \quad (3)$$

Where, U $_{ij}$ is between 0 and 1;
c_i is the centroids of cluster I;
d_{ij} is the Euclidean Distance between ith. Centroids c_i and j$_{th}$. Data point.
$m \in [1, \infty]$ is a weighting exponent. There is no prescribed manner for choosing the exponent parameter, "m". In practice, m=2 is common choice, which is equivalent to normalizing the coefficients linearly to make their sum equal to 1. When m is close to 1, then the cluster centre closest to the point is given much larger weight than the others and the algorithm is similar to K-Means.

To reach a minimum of dissimilarity function there are two conditions. These are given in (4) and (5).

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \quad (4)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \quad (5)$$

This algorithm determines the following steps in Figure2.

By iteratively updating the cluster centres and the membership grades for each data point, FCM iteratively moves the cluster centres to the "right" location within a data -set. FCM does not ensure that it converges to an optimal solution, because the cluster centers are randomly initialised. Though, the performance depends on initial centroids, there are two ways as described below for a robust approach in this regard.

1) Using an algorithm to determine all of the centroids.
2) Run FCM several times each starting with different initial centroids.

More mathematical details about the objective function based clustering algorithms can be found in [10].

4. EXPERIMENTAL SETUP AND RESULTS

In this experiment, we have used a standard dataset, the raw data used by the KDD Cup 1999 intrusion detection contest [2].
However, in our experiment, we have used 10% KDD Cup'99 datasets. This database includes a variety of intrusions simulated in a military network environment that is common benchmark for evaluation of intrusion detection techniques. This data set consists of 65525 data instances, with 21 training attack types, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusion and is labelled as either normal or a certain attack type. The distribution of attacks in the KDDCup'99 dataset is highly unbalanced. Some attacks are represented with only

a few examples, e.g. the phf and ftp_write attacks, whereas the Smurf and Neptune attacks cover many records. In general, the distribution of attacks is dominated by probes and DoS attacks.

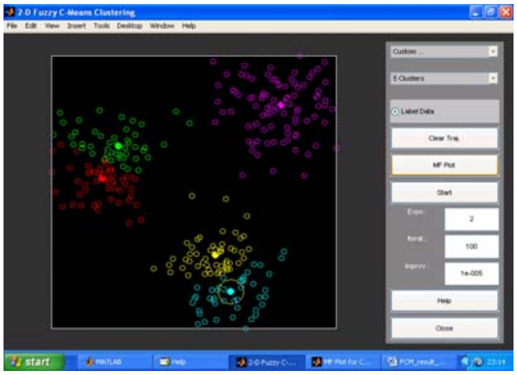
We carried out the experiments on 2.8GHz Pentium IV processor, 512 MB RAM running Windows XP system. Fuzzy Logic Toolbox [11] of MATLAB 7.0 was used for fuzzy c-Means clustering.

In practice, the number of classes is not always known beforehand. There is no general theoretical solution to find the optimal number of clusters for any given dataset. We choose K=5 for the experimentation. The simulation results after using FCM are shown in figures 3, 4. In figure 5, the shape of the membership function for selected values of the fuzzification factor (m=2) and cluster number=5 is shown. It can also be seen from these figures that, we are able to group the data by using the objective functions based fuzzy c-means clustering approach. Finally, the relationships of the objective function with the number of iterations are obtained in figure6. Apart from all these, there are standard measures for evaluating IDSs include detection rate, false positive rate, Recall rate, ROC (Receiver Operating characteristics). These are good indicators of performance, since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications are made in the process. Based on this, ROC for K-Means clustering is obtained as a measure of the effectiveness of the system, which is shown in figure7. The time taken to design the model is varying from 11 seconds to 2 minutes as the number of cluster increases from 2 to 10.

FCM ALGORITHM:

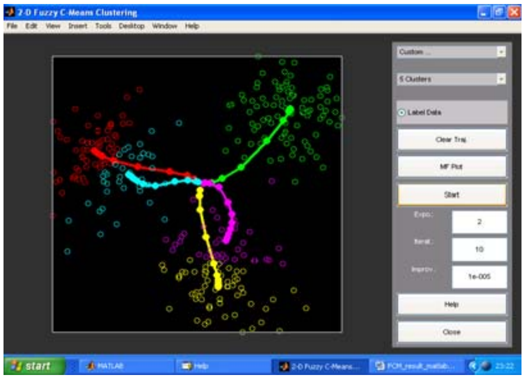
Input: n data objects, number of clusters
Output: membership value of each object in each cluster
Algorithm:
1. Select the initial location for the cluster centres
2. Generate a new partition of the data by assigning each data point to its closest centre.
3. Calculate the membership value of each object in each cluster.
4. Calculate new cluster centers as the centroids of the clusters.
5. If the cluster partition is stable then stop, otherwise go to step2 above.

Figure2. Fuzzy c-Means Clustering Algorithm



Custom ...
5 Clusters
Expo. 2
Iterat. 100
Improv. 1e-005

Figure3. Five clusters of data after using FCM



Custom ...
5 Clusters
Expo. 2
Iterat. 10
Improv. 1e-005

Figure4. 2-D Fuzzy c-Means Clustering

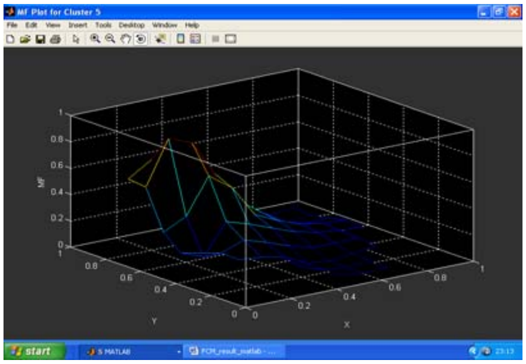


Figure5. MF (membership function) Plot for FCM

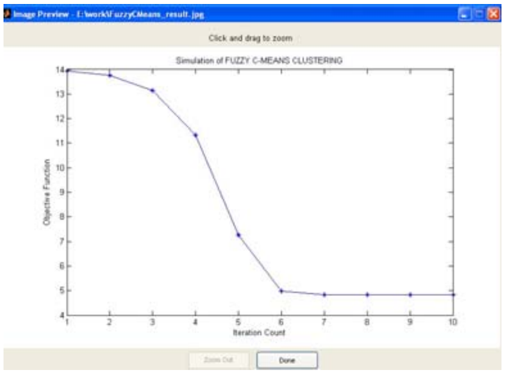


Figure6. Simulation of FCM

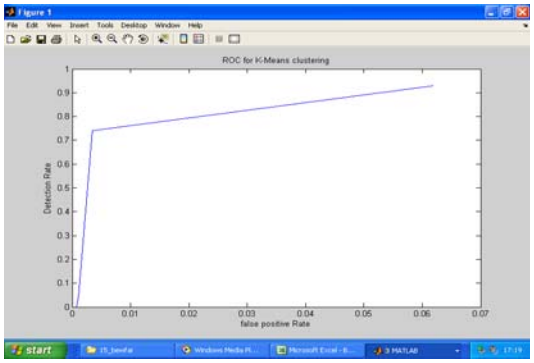


Figure7. Receiver Operating Characteristic (ROC) for K-Means Clustering

5. DISCUSSION

Though the K-Means algorithm is popular for its simplicity, it has some drawback in choosing optimal number of clusters. However, the fuzzy based clustering methods had shown tremendous achievements in areas of image processing and pattern recognition, and intrusion detection. The fuzzy c-means is a good choice for circular and spherical clusters, but if the orientation of natural clusters is not spherical, then the algorithm leads to among almost wrong clusters. Another drawback of the algorithm is that it imposes equal size clusters on the data set which is again a deviation from the natural clusters. The performance of any fuzzy based clustering method is the best when the number of clusters is known Apriori. But most of the time, it is not the case and so researchers have devised a number of methods known as cluster validation indices to evaluate the clusters formed [12, 13].

6. RELATED WORK

In [14], a speed up technique for image data was proposed. In this method, FCM convergence is obtained by using a data reduction method. Data reduction is done by quantization and speed-up by aggregating similar examples, which were then represented by a single weighted exemplar. The objective function of the FCM algorithm was modified to take into account the weights of the exemplars. However, the presence of similar examples might not be common in all data sets. They showed that it performs well on image data sets. However, the above algorithm does not address the issue of clustering large or very large datasets under the constraints of limited memory.

Recently in [15], a sampling based method has been proposed for extending fuzzy and probabilistic clustering to large or very large data sets. The approach is based on progressive sampling, which can handle the non-image data. However, the termination criteria for progressive sampling could be complicated as it depends upon the features of the data sets.

In [16], two methods of scaling EM to large data sets have been proposed by reducing time spent in E-step. In the first method, which is referred to as incremental EM, data is partitioned into blocks and then incrementally updating the log-likelihoods. In the second method, lazy EM, at scheduled iterations the algorithm performs partial E and M steps on a subset of data. The methods used to scale EM may not generalize to FCM as they are different algorithms with different objective functions.

7. CONCLUSION

The applications of fuzzy based methods in all fields of engineering and sciences have shown far reaching results and their applications in intrusion detection are also optimistic. In this paper, we have discussed the objective function based fuzzy c-means clustering in detail and their application in detecting anomaly based network intrusions. Fuzzy clustering leads to information granulation in terms of fuzzy sets or fuzzy relations. Membership grades are important indicators of the typicality of patterns or their borderline character. The advantage of using fuzzy logic is that it allows one to represent concepts where objects can fall into more than one category (or from another point of view- it allows representation of overlapping categories). The results obtained in this paper show that FCM works very efficiently in obtaining compact well separated clusters to detect network intrusions. Though we have already seen many examples of successful application of cluster analysis, there still remain many open problems due to the existence of many uncertain factors. These problems have already attracted and will continue to attract intensive efforts from broad discipline. However, a major problem with fuzzy clustering is that it is difficult to obtain the membership values. A general approach may not work because of the subjective nature of clustering. It is required to represent clusters obtained in a suitable form to help the decision maker. Knowledge-based clustering schemes generate intuitively appealing descriptions of clusters. They can be used even when the patterns are represented using a combination of qualitative and quantitative features, provided that knowledge linking a concept and the mixed features are available. However, implementations of the conceptual clustering schemes are computationally expensive and are not suitable for grouping large data sets. The K-means algorithm is most successfully used on large data sets. This is because K-means algorithm is simple to implement and computationally attractive because of its linear time complexity. However, it is not feasible to use even this linear time algorithm on large data sets. In summary, clustering is an interesting, useful, and challenging problem. It has great potential in applications like object recognition, image segmentation, anomaly detection and information filtering and retrieval. However, it is possible to exploit this potential only after making several designs choices carefully.

REFERENCES

[1] J.Allen, A. Christie, W.Fithen, j.McHugh, J.pickel, and E.Stoner, "State of the practice of Intrusion Detection Technologies", CMU/SEI-99-TR-028, Carnegie Mellon Software Engg. Institute. 2000.

[2]KDDCup'1999dataset. http://kdd.ics.uci.edu/databases/kddcup'99/kddcup99.html.

[3]S.theodoridis and K.koutroubas, "pattern Recognition", Academic Press, 1999.

[4]Wikipedia-Cluster Analysis, http://en.wikipedia.org/wiki/cluster_analysis.

[5]Johan Zeb Shah and anomie bt Salim, "Fuzzy clustering algorithms and their application to chemical datasets", in Proc. Of the post graduate Annual Research seminar 2005, pp.36-40.

[6]Zhengxim Chen, "Data Mining and Uncertain Reasoning-An integrated approach", Willey, 2001.

[7]Witcha Chimphlee, et.al. "Un-supervised clustering methods for identifying Rare Events in Anomaly detection", in Proc. Of World Academy of Science, Engg. and Tech (PWASET), Vol.8, Oct2005, pp.253-258.

[8]J.Bezkek, "pattern Recognition with fuzzy objective function algorithms", Plennum Press, USA, 1981.

[9]S.Albayrak, and Fatih Amasyali, "Fuzzy C-Means clustering on medical diagnostic systems", International XII Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN-2003.

[10]Wit old Pedrycz, "Knowledge Based Clustering", John Willey&sons Inc., 2005.ISBN:0-471-46966-1.

[11]MATLAB 7.0, Math Works, statistical Toolbox. www.mathworks.org.

[12]V.Maulik, S.Bandopadhyay, "Performance evaluation of some clustering Algorithms and Validity indices", IEEE transaction on Pattern Analysis and Machine Intelligence, vol.24, no.12, pp.1650-1654, Dec.2002.

[13]Steven Eschrich, Jingwei Ke, Lawrence o. Hall, and Dmitry B. Goldgof, "Fast accurate fuzzy Clustering through data reduction",IEEE Transaction on Fuzzy Systems. Vol.11, 2, pp.262-270, 2003.

[14]Richard J. hathaway and james C. Bezdek, "Extending fuzzy and probabilistic clustering to very large datasets", Journal of Computational statistics and data analysis, vol.51, issue1, pp.215-234, Nov.2006.

[15]Bo Thiesson, Christopher Meek, and david hackerman, "Accelerating EM for large Database", Machine Learning Journal, v.45, pp.279-299, 2001.

[16] A K Jain and R C Dubes, "Algorithm for Clustering Data", Prentice Hall, Engle Wood cliffs, NJ, USA, 1988.