An Efficient Clustering and Distance Based Approach for Outlier Detection

Garima Singh[1], Vijay Kumar[2]

[1]*M.Tech Scholar, Department of CSE, MIET, Meerut, Uttar Pradesh, India*

[2]*Assistant Professor, Department of CSE, MIET, Meerut, Uttar Pradesh, India*

Abstract— **Outlier detection is a substantial research problem in the domain of data mining that aims to uncover objects which exhibit significantly different, exceptional and inconsistent from rest of the data. Outlier detection has been widely researched and finds use within various application domains including tax fraud detection, network robustness analysis, network intrusion and medical diagnosis. In this paper we propose an efficient clustering and distance based outlier detection technique. The clustering algorithms employed for this task are PAM, CLARA and CLARANS and a novel clustering algorithm I-CLARANS is proposed. The process of outlier detection is divided into two stages. In the first stage clustering is performed and in the second stage outlier detection is performed. The purpose is to perform clustering and outlier mining simultaneously. The experimental results depict that the proposed method is effective and promising in practice. We also present comparison of proposed algorithm with existing algorithms to validate its advantage in outlier detection.**

Keywords— **Outlier detection, Data Mining, Clustering, PAM, CLARA, CLARANS.**

I. INTRODUCTION

With the rapid development of computer and information technology in the last decades, gigantic amounts of data are constantly being generated and collected, and data mining and knowledge discovery becomes the essential scientific discovery process. Data mining is the efficient discovery of valuable, non obvious information from a large collection of data. It is a process that helps identify new opportunities by finding fundamental truths in apparently random data. The patterns revealed can shed light on application problems and assist in more useful, proactive decision making [1]. Extensive research in data mining has been done on discovering patterns about the underlying data. The techniques of pattern discovering include classification, association, outlier and clustering.

Outlier detection is considered pivotal in numerous research areas and wide variety of application domains. In many data analysis tasks we need to record and sample a large number of variables. Identification of outliers is the prominent activity which facilitates coherent analysis. The failure to detect or incorrect treatment of outliers may otherwise adversely lead to model misspecification, biased parameter estimation and incorrect results. Hence it is crucial to detect them prior to modeling and analysis. Outlier detection methods have been suggested for various applications, such as credit card fraud detection, clinical trials, voting irregularity analysis, data cleansing, network intrusion, severe weather prediction, geographic information systems, athlete performance analysis, and other data-mining tasks [2].

Many approaches to detect outliers have been studied in data mining community. These approaches can be classified into Statistical model-based approaches, Depth-based approaches, Clustering-based approaches, Distance based approaches and Density-based approaches. In the past decade there has been intensive research on clustering based outlier detection methods, which have the advantage of simple modeling and effectiveness. Clustering methods aim to find clusters. As a result, they optimize clustering not outlier detection. To address this issue, recently various approaches for outlier detection have been merged together.

In this paper we propose an outlier detection technique which is a combination of partition clustering algorithm and distance-based outlier detection method. Firstly we introduce a partitioning based clustering algorithm that groups the data having similar features. Then we apply distance based outlier detection method to detect the outliers. We aim to perform process of clustering and outlier detection simultaneously.

The paper is organized as follows. Section II provides a brief discussion on the previous works related to the concerned topic. Section III explains working of partition based clustering algorithms PAM, CLARA and CLARANS. Section IV presents proposed algorithm I-CLARANS. The proposed methodology is presented in Section V, while Section VI presents the experimental results and analysis. Conclusion and future research directions are provided in Section VII.

II. RELATED STUDIES

Outlier detection is also referred to as anomaly detection, event detection, novelty detection, deviant discovery, change point detection, fault detection, intrusion detection or misuse detection. A variety of supervised, semi-supervised and unsupervised techniques have been used for outlier detection and each has their own strengths and weaknesses. An approach for discovering outliers using distance metrics was proposed by [4], [5]. They define a point to be a distance outlier if at least a user-defined fraction of the points in the data set are further away than some user-defined minimum distance from that point. In their experiments, they primarily focus on data sets containing only continuous attributes. Points that do not cluster well are labeled as outliers [6].

One approach is that of statistical model-based outlier detection, where the data is assumed to follow a parametric (typically univariate) distribution [7]. A Gaussian mixture model was proposed by [8]. Where each data point is given a formulated score and data point which have a high score declared as outlier. Detecting outlier based on the general pattern within data points was proposed by [9] where it combines a Gaussian mixture model and supervised method. This approach is can also referred as parametric approach.

Depth based outlier detection [10] is one the variant of statistical outlier detection where each data object of dataset represented by n-d space having a assigned depth. These data points are organized into convex hull layers according to assigned depth and outlier is formulated on the basis of shallow depth values. These models are not suitable for high dimensional data set.

The clustering based techniques involve a clustering step which partitions the data into groups which contain similar objects. The assumed behavior of outliers is that they either do not belong to any cluster, or belong to very small clusters, or are forced to belong to a cluster where they are very different from other members [11]. A mining based method used for detecting outliers over categorical data using hyper graph model has also been proposed [12].

Recently, density-based approaches to outlier detection have been proposed [13]. In this approach, a local outlier factor (LOF) is computed for each point. The LOF of a point is based on the ratios of the local density of the area around the point and the local densities of its neighbors. The size of a neighborhood of a point is determined by the area containing a user-supplied minimum number of points (MinPts). A similar technique called LOCI (Local Correlation Integral) is presented in [14].A comparison of various anomaly detection schemes is presented in [15].

III. Partition Based Clustering Algorithms

The procedure followed by partitioning algorithms can be stated as follows: “Given n objects, these methods construct k partitions of the data, by assigning objects to groups, with each partition representing a cluster. Generally, each cluster must contain at least one object; and each object may belong to one and only one cluster, although this can be relaxed”. It is an iterative relocation technique is used to improve the clustering by moving up the object from one group to another. Partition based clustering is represent by centriod or medoid [16].They use iterative way to produce the clustering. This section provides the working behind PAM, CLARA, and CLARANS clustering algorithms for outlier detection.

A. Partitioning Around Medoid (PAM)

PAM is developed by Kaufman and Rousseuw in 1987. The algorithm chooses k- medoid initially and then swaps the medoid object with non medoid as a result quality of cluster is improved. The PAM algorithm forms clusters by examining all objects that are not medoids. This imposes an expensive computation cost of $O(k(n-k)^2)$ in each iteration [17]. Algorithm work well with small dataset but does not work well with large dataset. However it is very robust when compare with k-mean in the presence of noise or outlier. PAM procedure is given in Figure 1, where k is the number of clusters, n is the number of objects in the datasets, S is the set of objects to be clustered, s_j is an object $\in$ S, R denotes the set of objects $\in$ S selected of medoids, r_j, $r_c \in$ R, d is the dissimilarity function.

Algorithm PAM

1. Build Phase: Randomly select two initial data points as medoids. The selection is made in such a way that the dissimilarity to all other data objects is minimal. The main objective of this step is to decrease the objective function.
2. Swap Phase: The Swap phase computes the total cost ‘T’ for all pairs of objects r_i and s_h, where $r_i \in$ R is currently selected and $s_h \in$ S is not.
3. Selection Phase: This phase selects the pair (r_i, s_h) which minimizes ‘T’. If the minimum T is negative, the swap is carried out and the algorithm reiterates Step 2. Otherwise, for each non-selected object, the most similar medoid is found and the algorithm stops.

B. Clustering LARge Applications (CLARA)

To deal with larger data sets, a sampling-based method, called Clustering LARge Applications (CLARA) was developed by Kaufman and Rousseeuw (1990). CLARA draws multiple samples of the data set, applies PAM on each sample, and returns its best clustering as the output. The complexity of each iteration now becomes $O(ks^2 + k(n-k))$, where s is the size of the sample, k the number of clusters, and depends on the sampling method and the sample size. CLARA cannot find the best clustering if any sampled medoid is not among the best k medoids. When the sample size is small, CLARA’s efficiency in clustering large data sets comes at the cost of clustering quality. Therefore, it is difficult to determine the sample size. Experiments provided in [18] indicate that 5 samples of size 40 + 2k gave satisfactory results. However, this is only valid for a small k.

```
Algorithm CLARA
1. For i=1 to 5, repeat Steps 2 to 5.
2. Draw a sample of 40 + 2k objects randomly from the entire data set and call PAM algorithm to find k medoids of the sample.
3. For each object O in the entire data set, determine k-medoids which is most similar to O.
4. Calculate average dissimilarity of the clusters obtained from Step 3. If this value is less than current minimum, use the new value as current minimum and retain the k medoids found in Step 2 as the best set of medoids obtained so far.
5. Return to Step 1 to start the next iteration.
```

C. Clustering large Applications based upon RANdomized Search(CLARANS)

CLARANS [16] algorithm mix both PAM and CLARA by searching only the subset of the dataset and it does not confine itself to any sample at any given time. One key difference between CLARANS and PAM is that the former only checks a sample of the neighbors of a node. But, unlike CLARA, each sample is drawn dynamically in the sense that no nodes corresponding to particular objects are eliminated outright. In other words, while CLARA draws a sample of nodes at the beginning of a search, CLARANS draws a sample of neighbors in each step of a search. This has the benefit of not confining a search to a localized area.

```
Algorithm CLARANS
1. Input parameters numlocal and maxneighbor. Initialize i to 1, and mincost to a large number.
2. Set current to an arbitrary node in Gn;k.
3. Set j to 1.
4. Consider a random neighbor S of current, and based on S, calculate the cost differential of the two nodes.
5. If S has a lower cost, set current to S, and go to Step 3.
6. Otherwise, increment j by 1. If j maxneighbor, go to Step 4.
7. Otherwise, when j > maxneighbor, compare cost of the current with mincost. If the former is less than mincost, set mincost to the cost of current and set bestnode to current.
8. Increment i by 1. If i > numlocal, output bestnode and halt. Otherwise, go to Step 2.
```

Steps 3 to 6 above search for nodes with progressively lower costs. But, if the current node has already been compared with the maximum number of the neighbors of the node (specified by maxneighbor) and is still of the lowest cost, the current node is declared to be a “local” minimum. Then, in Step 7, the cost of this local minimum is compared with the lowest cost obtained so far. The lower of the two costs above is stored in mincost. Algorithm CLARANS then repeats to search for other local minima, until numlocal of them has been found. As shown above, CLARANS has two parameters: the maximum number of neighbors examined (maxneighbor) and the number of local minima obtained (numlocal). The higher the value of maxneighbor, the closer is CLARANS to PAM, and the longer is each search of a local minima. But, the quality of such a local minima is higher and fewer local minima need to be obtained. Algorithm CLARANS then repeats to search for other local minima, until numlocal of them have been found. The computational complexity is $O(N^2)$ where N is the number of objects [17].

IV. Proposed Algorithm

Improved CLARANS (I-CLARANS) is a new partitioning based clustering algorithm which aims to improve CLARANS. I-CLARANS is a novel method to find the best possible medoid set in few iterations. Our approach is to find best medoid using the geometric properties which facilitate estimation of that region of the dataset where probability of finding the medoid is highest. Unlike methods based on sampling, our technique takes into consideration all elements, thereby increasing the chances of making a good (if not optimal) choice for the medoid element.

```
Algorithm I-CLARANS
1. Input parameters numlocal and maxneighbour. Initialize i to 1, and mincost to a large number.
2. Randomly choose an element e from dataset D.
3. Find farthest element e1 from e.
4. Find farthest element e2 from e1.
5. Find median element $e_m$ as the one that splits elements such that half are nearer to e1 and half are nearer to e2.
6. Set j to 1.
7. Consider a random neighbor S of median $e_m$, calculate the cost differential of the two nodes.
8. If S has a lower cost, set median element to S, and go to Step 6.
9. Otherwise, increment j by 1. If j maxneighbour, go to Step 7.
10. Otherwise, when j > maxneighbour, compare the cost of median element with mincost. If the former is less than mincost, set mincost to the cost of median element and set best node to median element.
11. Increment i by 1. If i > numlocal, output best node and halt. Otherwise, go to Step 5.
```

V. Proposed Methodology

We have divided the process of outlier detection divided into two modules. In the first module clustering is performed using one the four clustering algorithms and in the second stage outlier detection is performed. The main purpose is to simultaneously perform clustering and outlier detection. The proposed methodology is shown in Fig. 1.

Clustering Module: In the first module partition clustering is performed using one of the algorithms PAM/ CLARA/ CLARANS/ I-CLARANS. The algorithm produces a set of clusters and a set of medoids (cluster centers). Small clusters are regarded as outliers and removed from the dataset. To define small clusters [19] approach is used It defines small cluster as a cluster with fewer points than half the average number of points in the k number of clusters. These algorithms reduce the processing time as well as the size of data set to be processed by partitioning the data set into groups which consist of similar data points. Use of partition based clustering enables processing of outlying data elements instead of individual data points.

Outlier Detection Module: The second module aims to detect outliers. To find outliers in large clusters, we compute the Absolute Distances between the Medoid, μ, of the current cluster and each one of the Points, pi, in the same cluster. The produced value will be termed (ADMP). If the ADMP value is greater than a calculated threshold, T, then the point is considered an outlier; otherwise, it is not. The value of T is calculated as the average of all ADMP values of the same cluster multiplied by 1.5.

$$ADMP = | pi - \mu | \qquad (1)$$

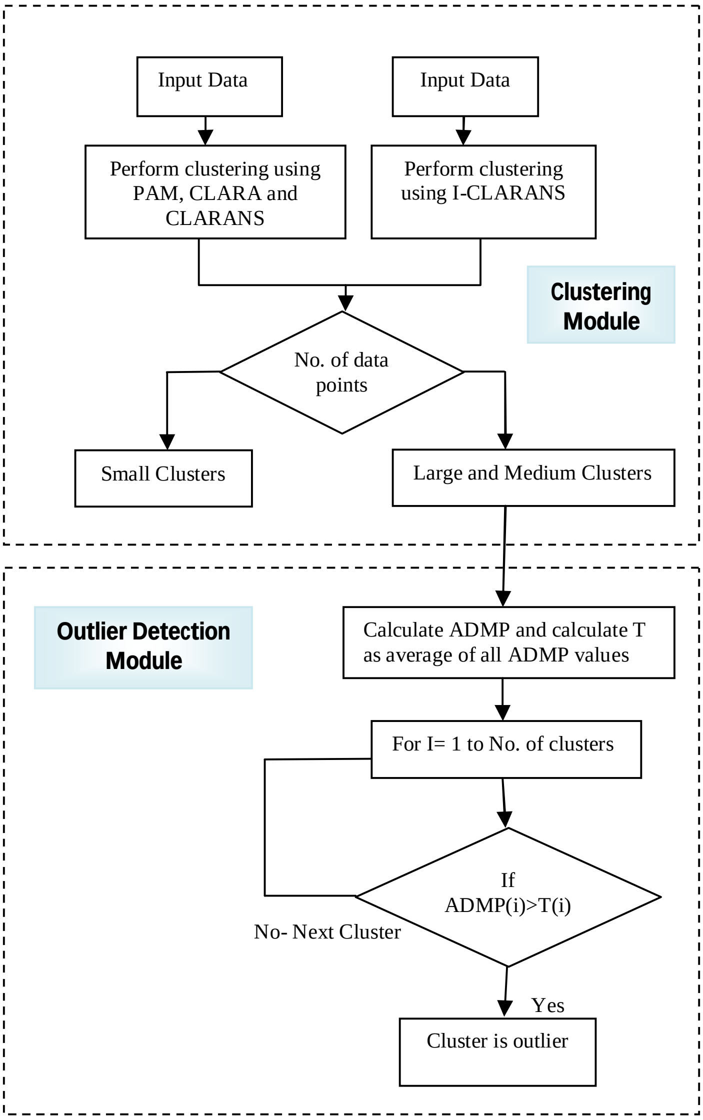
Input Data
Input Data
Perform clustering using PAM, CLARA and CLARANS
Perform clustering using I-CLARANS
Clustering Module
No. of data points
Small Clusters
Large and Medium Clusters
Outlier Detection Module
Calculate ADMP and calculate T as average of all ADMP values
For I= 1 to No. of clusters
If ADMP(i)>T(i)
No- Next Cluster
Yes
Cluster is outlier

Fig. 1 Flow diagram of proposed methodology

VI. Experimental Results and Analysis

All of our experiments are conducted on a laptop with Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz and 6.00 GB RAM, running Windows 7 Home Premium operating system. The algorithms are implemented in Microsoft Visual C++ 2010 Express on two numeric value datasets Fisher's Iris dataset and Bupa data set. Iris dataset comprises of 150 records, 3 classes and 4 dimensions. Bupa data set comprises of 345 records, 2 classes and 6 dimensions. Both the datasets are taken from UCI Machine Learning Repository. (http://archive.ics.uci.edu/ml/datasets). Table I shows the comparison between the existing clustering algorithms PAM, CLARA, CLARANS and the proposed clustering algorithm I-CLARANS in terms of outlier accuracy.

TABLE I
Number Of Outliers Detected

Dataset	PAM	CLARA	CLARANS	I-CLARANS
IRIS	17	18	18	19
BUPA (Class 1)	14	17	19	20
BUPA (Class 2)	18	21	22	22

It could be seen that the I-CLARANS method is better when compared with PAM, CLARA and CLARANS. The performance of both CLARA and CLARANS is same while using Iris dataset. With BUPA dataset PAM algorithm detected 32 outliers, CLARA detected 38 outliers, CLARANS found 41 outliers and I-CLARANS found 42 outliers respectively. Thus, it could be deduced that the I-CLARANS algorithm is efficiently detecting the outliers from both the datasets. The chart below represents the obtained results graphically.

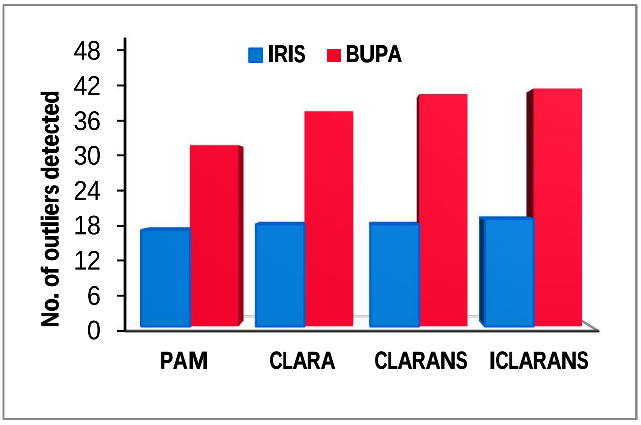

Fig. 2 Number Of Outliers Detected

In our next set of experiments we compare the algorithms on the basis of their time complexity i.e. time taken by them to detect the outliers. Table II shows the experimental results.

TABLE III
TIME COMPLEXITY

Dataset	PAM	CLARA	CLARANS	I-CLARANS
IRIS	5.12	5.36	4.14	3.72
BUPA	8.62	6.77	5.79	4.93

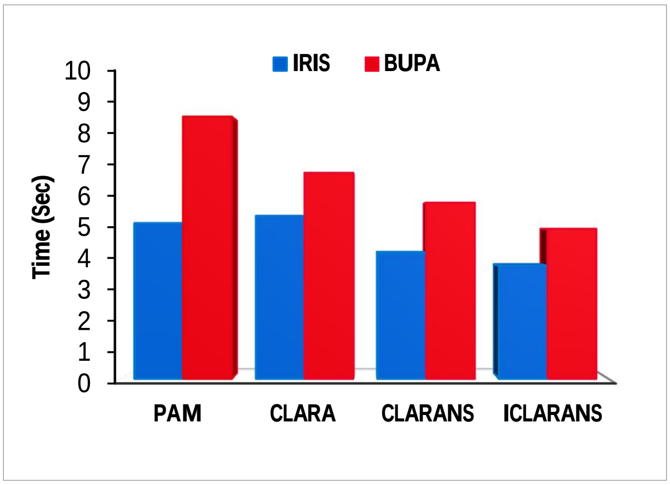

Fig. 3 Time Complexity

For small datasets, CLARANS outperforms PAM considerably. Also, CLARANS is more efficient than PAM and CLARA for small and large data sets [17]. It is evident from TABLE II that the fastest algorithm is I-CLARANS, followed by CLARANS, CLARA AND PAM.

VII. CONCLUSIONS

In this paper we took into consideration three existing partition based clustering algorithms PAM, CLARA and CLARANS and proposed a novel clustering algorithm called I-CLARANS. We combined these algorithms with distance based method for outlier detection. The clustering algorithms facilitate grouping of data with similar characteristics into clusters thereby reducing size of datasets. This reduces computation time considerably. Then ADMP measure is considered to finally detect the outliers. We analyzed the performance of our proposed algorithm I-CLARANS against existing algorithms in terms of outlier accuracy and time complexity. The experimental results on various datasets show that the proposed algorithm I-CLARANS identifies outliers more successfully than existing algorithms.

The main contribution of this paper is design of an efficient outlier detection approach utilizing partition base clustering algorithms and distance method. The approach needs to be implemented on more complex and varying datasets. Future work requires approach applicable for categorical and high dimensional datasets.

REFERENCES

[1] Bigus Joseph P., *Data Mining with Neural Networks*, McGraw–Hill, U.S.A., 1996

[2] Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L.A comparative study of rnn for outlier detection in data mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*. IEEE Computer Society, USA, 2002

[3] J. J. Han, and M. Kamber, “Data Mining Concepts and Techniques,” *Morgan Kaufmann*, USA, 2001.

[4] E. Knorr and *et al.* Distance-based outliers: Algorithms and applications. *VLDB Journal*, 2000.

[5] E. Knorr and R. Ng. A unified notion of outliers: Properties and computation. In *ACM SIGKDD*, 1997

[6] E. Knorr and R. Ng. Finding intentional knowledge of distance-based outliers. In *VLDB*, 1999.

[7] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley, 1994.

[8] K. Yamanishi and J. Takeuchi.On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *Proceedings of Data Min. Knowledge Discovery*. Vol. 8, No. 3, pp 275-300, 2004

[9] K. Yamanishi and J. Takeuchi, 2001. Discovering outlier filtering rules from unlabeled data-combining a supervised learner with an unsupervised learner. In *Proceedings of KDD*, pp 389-394, 2001

[10] R. Nuts and P. Rousseeuw. Computing depth contours of bivariate point clouds. *Computational Statistics and Data Analysis*, Vol 23, No 2, pp 153-168, 1996

[11] M. F. Jiang, S.S. Tseng, and C.M. Su. “Two-phase clustering process for outliers detection”, *Pattern Recognition Letters*, Vol 22, No. 6-7, pp. 691-700, 2001

[12] Zengyou He, Shengchun Deng , Xiaofei Xu. Outlier detection integrating semantic knowledge. In *Proceedings of Third international Conference on Advances in Web-Age information Management*., Vol. 2419. pp 126-131. 2002.

[13] Markus M. Breunig, Hans-Peter Kriegel, Raymond T.Ng, and Jorg Sander. LOF: Identifying density-based local outliers. In *ACM SIGMOD International Conference on Management of Data*, 2000.

[14] Spiros Papadimitriou, Hiroyuki Kitawaga, PhillipB. Gibbons, and Christos Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *ICDE*,2003.

[15] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of outlier detection schemes for network intrusion detection. In *SIAM Data Mining*, 2003.

[16] Shalini S Singh, N C Chauhan, “K-means v/s Kmedoids: A Comparative Study”, *National Conference on Recent Trends in Engineering & Technology*, May 2011.

[17] Krzysztof Koperski, Junas Adhikary and Jiawei Han. Spatial Data Mining: Progress and Challenges Survey Paper, School of Computer Science Simon Fraser University Burnaby, 1996.

[18] Ng, R. and Han, J. “Efficient and Effective Clustering Methods for Spatial Data Mining” . In *Proceedings of 20th Conference. Very Large Databases*, Pp. 144–155,1994.

[19] A.Loureiro, L.Torgo and C.Soares, “Outlier detection using Clustering methods: A data cleaning Application”, In *Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector*. Bonn, Germany, 2004.

[20] Rousseeuw, P. and A. Leroy, Robust *Regression and Outlier Detection*, 3rd ed., John Wiley & Sons, 1996.

[21] Chandola, V., Banerjee, A. and Kumar, V. “Anomaly detection: A survey”, *ACM Computing Surveys*, Vol. 41, Issue 3, Pp.1-58, 2009

[22] P. Murugavel, Dr. M. Punithavalli, “Improved Hybrid Clustering and Distance-based Technique for Outlier Removal”, *International Journal on Computer Science and Engineering*, Volume 3, pp.333-339, 2011.

[23] S.Vijayarani, S.Nithya, “Sensitive Outlier Protection in Privacy Preserving Data Mining”, *International Journal of Computer Applications*, Volume 33, pp 19-27, 2011.

[24] Moh’d Belal Al-Zoubi, “An Effective Clustering-Based Approach for Outlier Detection”, *European Journal of Scientific Research*, Vol.28 No.2, pp. 310-316, 2009.

[25] R. T. Ng and J. Han CLARANS: A method for clustering objects for spatial data mining, IEEE Transactions on Knowledge and Data Engineering, 14 pp. 1003–1016, 2002.

[26] P Chandore, P Chatur.“Outlier Detection Techniques over Streaming Data in Data Mining: A Research Perspective”. *International Journal Journal of Advanced Research in Computer Science and Software Engineering*, Vol2, pp 12-16, June 2012.

[27] Pamula, R., Deka, J.K., Nandi, S. “An Outlier Detection Method Based on Clustering”. Emerging Applications of Information Technology, pp. 253–256, 2011

[28] Ms. S. D. Pachgade, Ms. S. S. Dhande. “Outlier Detection over Data Set Using Cluster-Based and Distance-Based Approach”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol 2, pp 12-16, June 2012.

[29] H M Koupaie, S. Ibrahim and J. Hosseinkhani.“Outlier Detection in Stream Data by Clustering Method”. International Journal of Advanced Computer Science and Information Technology,Vol. 2, No. 3,pp. 25-34, 2013.

[30] S. Salvador and P. Chan, “Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms”, in *Proceedings Sixteenth IEEE International Conference on Tools with Artificial Intelligence, Los Alamitos, CA, USA, IEEE Computer Society*, pp. 576–584 , 2004.

[31] K.Yoon, O.Kwon and D.Bae, “An approach to outlier Detection of Software Measurement Data using the Kmeans Clustering Method”, *First International Symposium on Empirical Software Engineering and Measurement, Madrid*.,pp:443-445, 2007

[32] V. J. Hodge and J. Austin, “A Survey of Outlier Detection Methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2,pp. 85–126, 2004.

[33] L. Portnoy, E. Eskin, and S. Stolfo, “Intrusion Detection with Unlabeled Data using Clustering,” in Proceedings of the ACM CSS Workshop on Data Mining Applied to Security (DMSA) ,pp. 5–8, 2001

[34] Velmurugan, T. and Santhanam, T. “A survey of partition based clustering algorithms in data mining: An experimental approach”, *Information Technology Journal.*, Vol.10, pp. 478-484, 2011.

[35] Jiang, S. and An, Q. Clustering-Based Outlier Detection Method, *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 2, pp.429-433, 2008.