RESEARCH ARTICLE OPEN ACCESS

Clustering Aggregation Using Control Chart Technique

Yamini Chalasani[1], M. Vani Pujitha[2]
[1,2]Department of Computer Science & Engineering, V.R.Siddhartha Engineering College(Autonomous), Affiliated to JNTU Kakinada, KANURU, Vijayawada, Krishna (DT), Andhra Pradesh, India.

Abstract
Data clustering is a process of putting similar data into groups. Point-based clustering aggregation is applying aggregation algorithms to data points and then combining various clustering results. Applying clustering algorithms to data points increases the computational complexity and decreases the accuracy. Many existing clustering aggregation algorithms have a time complexity quadratic, cubic, or even exponential in the number of data points. Thus Data fragments are considered. A Data fragment is any subset of the data that is not split by any of the clustering results. Existing model gives high clustering error rate due to lack of preprocessing of outliers. In the proposed approach, data fragments are considered and Outlier detection techniques are employed for preprocessing of data. New clustering aggregation algorithm proposed includes the outlier detection technique and each disjoined set of fragments is clustered in parallel thus reducing the time complexity.
***Keywords**:* Clustering aggregation, point-based approach, fragment-based approach, data fragment, computational complexity.

I. INTRODUCTION

Clustering is an important data-mining technique used to find data segmentation and pattern information. Data clustering takes the collected data which has similar characteristics directly into same cluster and analyzes the partnership among these objects or points. Data clustering is not only just one data mining method but as well as a pre-process data using preprocessing algorithms, knowledge discovery and data collection . The trouble of detecting clusters of points in data is challenging as soon as the clusters are of different size, density and shape. Several of these issues become much more significant when the data is of very high dimensionality and when it provides noise and outliers. Hierarchical clustering generates a tree of clusters by splitting or merging each cluster one for each level until the desired numbers of clusters are generated [1]. This generated tree is often known as dendogram (hierarchical tree). These algorithms use top-down approach (divisive) or bottom-up(agglomerative) or conceptual clustering(cobweb) to construct the dendogram. Agglomerative clustering algorithms considers each document for being single cluster and repeatedly merges two clusters which get most similar within their pattern each and every step until a single cluster of every document is obtained. Divisive clustering, then again, commences with all documents as a single cluster and splits them until all clusters are singleton clusters. Cobweb incrementally organizes observations into a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node. Ou Wu et al. [2] proposed that clustering aggregation algorithms can also be applied to data fragments instead of data points. A data fragment is any subset of the data that is not split by any of the clustering results. As the number of data fragments is much less than the number of data points the computational complexity decreases. But this gives high clustering error rate due to lack of preprocessing of outliers. Thus we include outlier detection technique prior to applying aggregation algorithm on the data set. Hierarchical algorithm can be applied in parallel process for clustering data fragments which reduces the time complexity [3].

The paper is organized as follows: Section 2 discusses about Literature survey, Section 3 discusses about proposed methodology, Section 4 discusses about Algorithm, Section 5 discusses about Evaluation of Experimental results and Section 6 describes Conclusions and Future work.

II. LITERATURE SURVEY

Xi-xian niu et al. [4] proposed the Local Agglomerative Characteristic (LAC) algorithm which mainly focuses on the local agglomerative characteristic of the data objects. The Main idea of LAC clustering is that two objects have higher similarity if they have the k shared nearest neighbor and have the relative higher local agglomerative characteristic in local data objects area at the same time [5]. Local characteristic is reflected by Local Average Distance (LAD) and Local Maximum Distance (LMD). Both LAD and LMD can reflect the local area data distribution characteristic. First, LMD is taken as local dynamic threshold, through simple compare and computation can get the LMD, but it is sensitive to local data point’s distribution shape. For the limitation of LMD’s representative of local data characteristics, second, LAD reflection of local data objects is checked out.

Advantages of this technique are it eliminates noisy points using LAD threshold. Proposed similarity measure is not only limited to consider the direct nearest neighbors, but also can take into the neighbor's local distributed feature. It is relatively resistant to noise and can handle clusters of arbitrary shapes and sizes, can deal with clusters of different density and natural distribution characteristics.

Drawbacks are that LMD and LAD both can give representative in some degree, but LMD can show direction and shape information but not represent most data point's, and LAD can reflect most data objects relative distance but direction information lost.

Ying Peng Yongyi Ma et al. [6] proposed an algorithm for belief functions. The information carrier in Dempster-Shafer theory (DST) is belief function. Combination of belief functions is required for getting a fusion result [7], [8]. Combination is performed just on condition that belief functions are related to the same event. It is necessary to distinguish which belief functions are reporting on which event. Here agglomerative algorithm is used for clustering purpose. Belief distance is taken as the dissimilarity measure between two belief functions, so there is no need of transformation. And due to the utilization of agglomerative algorithm, there is no need to set cluster number in advance. After getting the hierarchical tree, cluster number by threshold value is determined. Agglomerative algorithm creates a multilevel hierarchy tree, where clusters at one level are jointed as clusters at the next high level.

Advantages of this technique are it overcomes the problem of indirect clustering for possible inequality of transformation. This approach allows constructing clusters within uncertain information.

Drawbacks are Clustering approach used in this system virtually based on comparison between two belief function, which may has problems of hidden conflict among beliefs in one cluster. Partitioning tree depends on the level wise threshold values which would take more time to construct.

Cheng-Hsien Tang et al. [3] proposed the Distributed Hierarchical Agglomerative clustering algorithm which divides the whole computation into several small tasks, distribute the tasks to message-passing processes, and merge the results to form a hierarchical cluster [9],[10].This clustering algorithm uses the reduced similarity matrix to sequentially create disjoined sets of highly related data items. Each disjoined set is clustered in parallel to form a hierarchical sub-tree. Finally, the algorithm computes the similarity scores among all the sub-trees to form a complete hierarchical tree. To justify whether a data item belongs to a disjoined set, the distance (similarity) between two disjoined sets are to be defined. The similarity matrix (distance matrix) is a matrix that stores the similarity scores of all pairs of data items. A naive computation strategy that can concurrently calculate the matrix is to process each row in parallel.

Advantages of this technique are it takes less time to construct clusters due to parallel process. Takes less overhead i.e., if one processor handles one row, the execution time should depend on the time for computing the last row because it has the most work to do. Parallel computing provides a good way to handle large data sets.

Drawbacks are the space complexity of a similarity matrix is O (n^2) given n data items. If outliers are out of interest, only a small portion of similarity matrix is used to construct a hierarchical tree. It suffers with data clusters size, shape and outliers.

Jaruloj Chongstitvataa et al. [11] proposed an Agglomerative clustering algorithm which uses the concept of Attraction and Distraction provides higher accuracy for iris and haberman data sets when compared to K-means algorithm. A cluster of data objects can form either a concave shape or a convex shape. This method uses the distance between clusters and the cluster size as parameters for clustering. Clusters of objects are formed by attraction and distraction. In this work, Euclidean distance is used as the measurement of dissimilarity between objects. The distance between a pair of clusters is measured by the distance between the closest pair of points in each cluster. Attraction indicates if two clusters can be merged, based on the number of similar objects between two clusters, compared to the size of the cluster. Distraction indicates if the merging of two clusters should be deferred, based on other possible merge. In this method, a cluster is considered too small to be a cluster by itself if it is smaller than the median of the size of all clusters. Each of these small clusters is merged with its nearest neighbor cluster [12], [13]. It is found that this algorithm yields better accuracy on some datasets.

Advantages of this technique are it overcomes the restriction of the cluster shape, the concepts of attraction and distraction is used in this system effectively. The overall accuracy of the proposed method is better than K-means algorithm.

Drawback is that it always performs for concave shape clusters and had Quadratic time complexity.

Rashid Naseem et al. [14] proposed an Agglomerative Clustering technique used for restructuring of the program using Binary Features [15], [16]. This uses the Complete Linkage (CL) algorithm with Jaccard similarity measure using binary features, to group the similar statements into a noncohesive structured program. Binary features just indicate the absence or presence of a feature. The correct translation of a program into group of statements makes cohesive procedures. A program that is incorrectly translated may result in more problems as compared to original one. The programs taken from source code written in structured languages are restructured. A similarity measure is applied to compute similarity between every pair of entities, resulting in a similarity matrix. After applying clustering algorithm, hierarchy of results obtained and can be shown with the help of tree like structure known as dendrogram.

Advantages of this technique are this approach has the added benefit that it is very simple to understand and implement. It uses binary features, just to indicate the presence and absence of features. Effectively identifies the program reconstruction and program structures and tokens information. This helps to translate a non cohesive procedure into cohesive procedures.

Drawbacks are it does not suit for non structure programs. It does not handle special characters and new keywords information. It does not work for non binary features.

The COBWEB algorithm was developed by machine learning researchers in the 1980s for clustering objects in a object-attribute data set. The COBWEB algorithm yields a clustering dendrogram called classification tree that characterizes each cluster with a probabilistic description. Cobweb generates hierarchical clustering [17], where clusters are described probabilistically.

Advantages and disadvantages of cobweb are COBWEB uses a heuristic evaluation measure called category utility to guide construction of the tree. It incrementally incorporates objects into a classification tree in order to get the highest category utility. And a new class can be created on the fly, which is one of big difference between COBWEB and K-means methods. COBWEB provides merging and splitting of classes based on category utility, this allows COBWEB to be able to do bidirectional search. For example, a merge can undo a previous split. While for K-means, the clustering [18] is usually unidirectional, which means the cluster of a point is determined by the distance to the cluster centre. It might be very sensitive to the outliers in the data. COBWEB has a number of limitations. First, it is based on the assumption that probability distributions on separate attributes are statistically independent of one another. This assumption is, however, not always true because correlation between attributes often exists. Moreover, the probability distribution representation of clusters makes it quite expensive to update and store the clusters.

III. PROPOSED METHODOLOGY

Clustering aggregation provides a method for improving the clustering robustness by combining various clustering results. Consensus clustering, also called aggregation of clustering (or partitions), refers to the situation in which a number of different (input) clusterings have been obtained for a particular dataset and it is desired to find a single (consensus) clustering which is a better fit in some sense than the existing clusterings. The outliers in the data set are removed by applying control chart based outlier detection as described below.

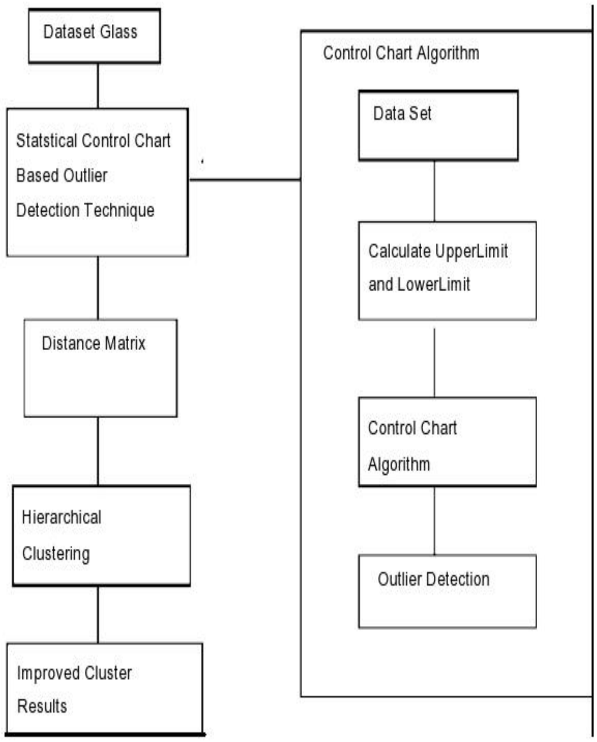

Figure1: Proposed Architecture

After the removal of outliers K-means clustering algorithm is applied on the data set. Each cluster generated by the K-means algorithm is termed as a fragment. These fragments are given as input to hierarchical clustering algorithm which gives the enhanced clustering results. Hierarchical clustering algorithms are of two types partitioning and conceptual. Here we are giving the results of conceptual hierarchical clustering that is cobweb. Partitioning clustering algorithms can also be applied i.e, agglometative(bottom-up) and divisive(top-down).

Control chart Technique(CCT): The purpose of a control chart is to detect any unwanted changes in the process. These changes will be signaled by abnormal (outlier) points on the graph. Basically, control chart consists of three basic components[19]:

1) A centre line, usually the mathematical average of all the samples plotted.
2) Upper and lower control limits that define the constraints of common cause variations.
3) Performance data plotted over time.

IV. ALGORITHMS

Steps included in the above architecture diagram is mentioned below :

Algorithm1: Outlierdetection(dataset)
Input : Glass dataset
Output: Glass dataset without outliers

1.1 Getatt(dataset)
1.2 Count(noofinstances)
1.3 For each instance in dataset
1.4 Calculate means for the instances
1.5 End for
1.6 Calculate standard deviation
1.7 Stddev(dataset)
1.8 Calculate UCL as mean+3sd
1.9 Calculate LCL as mean-3sd.
1.10 Calculate CL as 3sd
1.11 For each instance i in dataset
1.12 Find the range of i in LCL<=CL<=UCL
1.13 Find the outlier range
1.14 End for

Algorithm2: Algorithm Agglomerative

2.1 Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.

2.2 Find the least dissimilar pair of clusters in the current clustering,say pair
d[(r),(s)] = min d[(i),(j)]
where the minimum is over all pairs of clusters in the current clustering.

2.3 Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)]

2.4 Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:
d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]

2.5 If all objects are in one cluster, stop. Else, go to step 2

Algorithm3: Algorithm COBWEB

COBWEB(root, record):
Input: A COBWEB node root, an instance to insert record if root has no children then children := {copy(root)}
newcategory(record) \\ adds child with record's feature values.
insert(record, root) \\ update root's statistics else insert(record, root)
for child in root's children do calculate Category Utility for insert(record, child),
set best1, best2 children w. best CU. end for if newcategory(record) yields best CU then newcategory(record)
else if merge(best1, best2) yields best CU then merge(best1, best2)
COBWEB(root, record)
else if split(best1) yields best CU then split(best1)
COBWEB(root, record)
else COBWEB(best1, record end if end

Dataset Information:
Number of Instances: 214
Number of Attributes: 10 (including an Id#) plus the class attribute
all attributes are continuously valued
Attribute Information:
=> 1. Id number: 1 to 214
=> 2. RI: refractive index
=> 3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as
=> are attributes 4-10)
=> 4. Mg: Magnesium
=> 5. Al: Aluminum
=> 6. Si: Silicon
=> 7. K: Potassium
=> 8. Ca: Calcium
=> 9. Ba: Barium
=> 10. Fe: Iron
=> 11. Type of glass: (class attribute)
=> -- 1 building_windows_float_processed
=> -- 2 building_windows_non_float_processed
=> -- 3 vehicle_windows_float_processed
=> -- 4 vehicle_windows_non_float_processed (none in this database)
=> -- 5 containers
=> -- 6 tableware
=> -- 7 headlamps

Glass Data Basic Statistical Information:

Attribute:	Min	Max	Mean	SD	Correlation with class
% 2. RI:	1.5112	1.5339	1.5184	0.0030	-0.1642
% 3. Na:	10.73	17.38	13.407	0.8166	0.5030
% 4. Mg:	0	4.49	2.6845	1.4424	-0.7447
% 5. Al:	0.29	3.5	1.4449	0.4993	0.5988
% 6. Si:	69.81	75.41	72.6509	0.7745	0.1515
% 7. K:	0	6.21	0.4971	0.6522	-0.0100
% 8. Ca:	5.43	16.19	8.9570	1.4232	0.0007
% 9. Ba:	0	3.15	0.1750	0.4972	0.5751
% 10. Fe:	0	0.51	0.0570	0.0974	-0.1879

V. EVALUATION OF RESULTS

All experiments were performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2).
Figure 2 shows the experimental setup of netbeans IDE

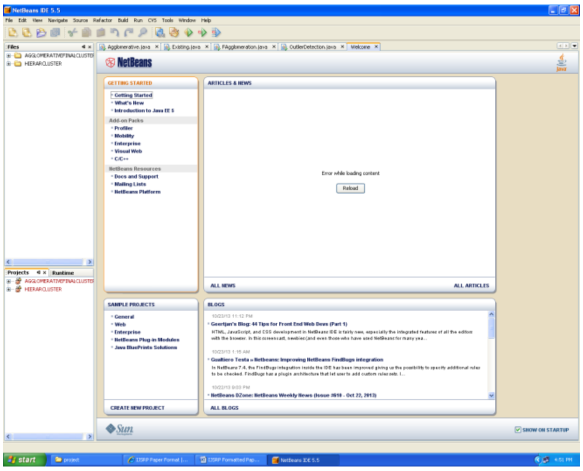

Figure 2: Experimental setup for netbeans IDE

Table 1 shows the instances of two datasets before and after applying the outlier detection technique

Table 1: Number of instances before and after applying outlier detection technique

ALGORITHM DATASET	BEFORE	AFTER
CPU	**208**	**185**
Glass	**214**	**178**

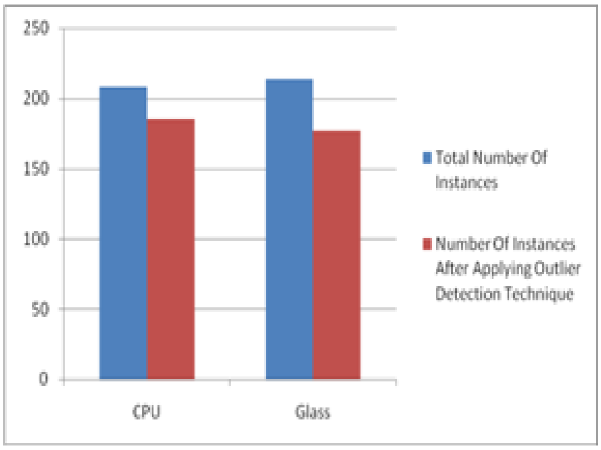

Figure 3:Total Number of Instances Before and After Applying CCT

Below figure shows the sum of squared error for K-means

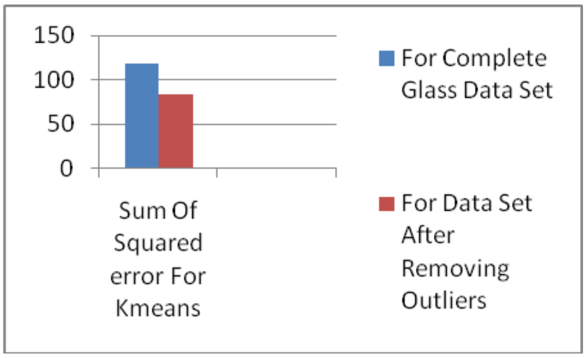

Figure 4:Sum of squared errors of k-means

Sample Fragment Instances:

Fragments Instances
2 4 (6%)
3 1 (2%)
5 1 (2%)
7 1 (2%)
8 50 (76%)
9 9 (14%)

Sample Dendogram:

FRAGMENT_NODE 0 [66]
| FRAGMENT_NODE 1 [8]
| | END(LEAF) 2 [4]Fragments are :1.51852,14.09,2.19,1.66,72.67,0,9.32,0,0,tableware,noFragments are :1.51829,14.46,2.24,1.62,72.38,0,9.26,0,0,tableware,noFragments are :1.51937,13.79,2.41,1.19,72.76,0,9.77,0,0,tableware,noFragments are :1.51905,14,2.39,1.56,72.37,0,9.57,0,0,tableware,no
| FRAGMENT_NODE 1 [8]
| | END(LEAF) 3 [1]Fragments are :1.52247,14.86,2.2,2.06,70.26,0.76,9.76,0,0,headlamps,no
| FRAGMENT_NODE 1 [8]
| | FRAGMENT_NODE 4 [2]
| | | END(LEAF) 5 [1]Fragments are :1.52177,13.75,1.01,1.36,72.19,0.33,11.14,0,0,'build wind non-float',no
| | FRAGMENT_NODE 4 [2]
| | | END(LEAF) 6 [1]Fragments are :1.51818,13.72,0,0.56,74.45,0,10.99,0,0,'build wind non-float',no

VI. CONCLUSIONS & FUTURE WORK

In the proposed work, data fragments are executed against data points and noisy free fragments are eliminated in order to improve the cluster accuracy. A data fragment is any subset of the data that is not split by any of the clustering results. As the number of data fragments is much less than the number of data points the computational complexity decreases. Proposed algorithm give better results. According to the rapidly changing technology new clustering algorithms are needed to decrease clustering error rate and increase the accuracy. Existing data mining clustering algorithms are very time consuming and they generate incorrect clusters, hence we extend fragment based hierarchical algorithm to detect outliers and remove them by adding outlier detection technique called CCT and then applying parallel hierarchical clustering algorithm to decrease time complexity.. Our future work can concentrate on testing this technique for various data sets and check for accuracy. Distance based outlier detection techniques can also be employed.

REFERENCES

[1] Aristides Giones ,"Clustering Aggregation" ,Yahoo! Research Labs, Barcelona Heikki Mannila University of Helsinki and Helsinki University of Technology.

[2] Ou Wu, Member, IEEE,WeimingHu, Senior Member, IEEE, Stephen J. Maybank, Senior Member, IEEE, Mingliang Zhu, and Bing Li " Efficient Clustering Aggregation Based on Data Fragments".

[3] Cheng-Hsien Tang, An-Ching Huang, Meng-Feng Tsai, Wei-Jen Wang "An Efficient Distributed Hierarchical-Clustering Algorithm for Large Scale Data", National Central University, Taiwan.

[4] Xi-xian Niu, Kui-he Yang, Dong Fu , "Local Agglomerative Characteristics based Clustering Algorithm", Hebei University of Science and Technology Shijiazhuang.

[5] Qian Weining, Zhou Aoying, Analyzing Popular Clustering Algorithms from Different Viewpoints. Journal of Software, 2002,13(8) :1382~1394.

[6] Ying Peng Yongyi Ma, Huairong Shen "Clustering Belief Functions Using Agglomerative Algorithm", Academy of Equipment Command & Technology Beijing, 101416, China.

[7] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," Annals of Mathematical Statistics, vol. 38, pp. 325–339.

[8] G. Shafer, "A Mathematical Theory of Evidence," Princeton: Princeton University Press.

[9] X. Li, "Parallel algorithms for hierarchical clustering and cluster validity," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 1088–1092, 1990.

[10] V. Olman, F. Mao, H. Wu, and Y. Xu, "Parallel clustering algorithm for large data sets with applications in bioinformatics," *IEEE/ACM* Transactions on Computational Biology and Bioinformatics, vol. 6, pp. 344–352, 2009.

[11] Jaruloj Chongstitvataa and Wanwara Thubtimdang "Clustering by Attraction and Distraction", Chulalongkom University, Bangkok 10330 THAILAND.

[12] B. Macqueen, "Some Methods for classification and Analysis of Multivariate Observations," Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley: University of California Press,pp. 281-297.

[13] R. C. Tryon, Cluster analysis. New York, McGraw-Hill.

[14] Rashid Naseem, Adeel Ahmed, Sajid Ullah Khan, Muhammad Saqib, Masood Habib "Program Restructuring Using Agglomerative Clustering Technique Based on Binary Features".

[15] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Transactionson neural networks, vol. 16, no. 3, pp. 645–678, 2005.

[16] L. Rokach, "A survey of Clustering Algorithms," Data Mining andKnowledge Discovery Handbook, pp. 269–298, 2010.

[17] Sanjoy Dasgupta ―Performance guarantees for hierarchical clustering‖ Department of Computer Science and Engineering University of California, San Diego

[18] V. Filkov and S. kiena. Integrating microarray data by consensus clustering. International Journal on Artificial Intelligence Tools, 13(4):863–880, 2004

[19] Zuriana Abu Bakar, "A Comparative Study for Outlier Detection Techniques in Data Mining".