Efficient Clustering of high dimensional nonlinear data using modified Denclue algorithm

R.NANDHAKUMAR [1] AND Dr. ANTONY SELVADOSS THANAMANI [2]

1. Research Scholar, Department of Computer Science ,NGM College,Pollachi-642001,India

2. Associate Professor and Head, Department of Computer Science, NGM College,Pollachi-642001,India

ABSTRACT

Clustering is a data mining task devoted to the automatic grouping of data, based on mutual similarity. Clustering in high-dimensional spaces is a recurrent problem in many domains. It affects time complexity, space complexity, Data Size Adaptability and Precision Value of clustering methods. High-dimensional data usually live in different low dimensional subspaces hidden in the original space. As high-dimensional objects appear almost alike, new approaches for clustering are required. Consequently, recent research has focused on developing techniques and clustering algorithms specifically for high-dimensional data.

In this research work, Denclue is chosen to analyse the compatibility of clustering high dimensional data. Denclue is a density based clustering technique. In density based clustering, the objects are classified based on their regions of density. These algorithms have the ability to discover classes of arbitrary shapes and omit noisy objects. To efficiently cluster the high dimensional nonlinear data, a new modified Denclue is proposed by incorporating the mathematics methods such as meta-heuristics, curse of dimensionality, sub spaces, data routing, correlation, normal distribution and darboux variate. The advantages of this proposed algorithm are it works on erroneous data, noisy data, provides better Clustering Pace, Competence Rate, Data Size Adaptability, Precision Value and Prognostic Reliability.

Keywords: Clustering, High dimensional data, Denclue, Mathematical methods.

1. Introduction

The main aspiration of clustering is to find high quality clusters within reasonable amount of time. Clustering in data mining is the process of discovering groups. Each group is a dataset such that the similarity among the data inside the group is maximized and the similarity in outside group is minimized. The discovered clusters are then used to explain the characteristics of the data distribution. Today there is tremendous necessity in

clustering the high dimensional data. For example, many business applications, clustering can be used to describe different customer groups and allows offering customized solutions. Clustering can be used to predict customer buying patterns based on their profiles to which cluster they belong.

Real-world datasets have very high dimensional feature space and is highly sparse. It becomes difficult to generate meaningful results from such redundant and sparse data through traditional clustering algorithm. This is due to the fact that when dimensionality increases, data becomes sparse since data points are located at different dimensional subspaces. Thus it requires greater computational power to compute clusters as distance measure to find similarities between data objects become meaningless and often noise becomes prevalent and masks the real cluster to be discovered.

The motivation of this research work is to enhance the high dimensional nonlinear data clustering method by proposing new algorithm called M-Denclue Algorithm. Mathematical methods such as meta heuristics, curse of dimensionality, data routing, correlation, normal distribution and Darboux variate are added with existing Denclue algorithms in order to efficiently cluster the high dimensional non linear data

2. Literature Survey

Alexander Hinneburg and Daniel A. Keim introduced a new algorithm to clustering in large multimedia databases called DENCLUE (DENsity-based CLUstEring). The basic idea of our new approach is to model the overall point density analytically as the sum of influence functions of the data points. Clusters can then be identified by determining density-attractors and clusters of arbitrary shape can be easily described by a simple equation based on the overall density function. The advantages of new approach are (1) it has a firm mathematical basis, (2) it has good clustering properties in data sets with large amounts of noise, (3) it allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets and (4) it is significantly faster than existing algorithms. To demonstrate the effectiveness and Competence Rate of DENCLUE, they perform a series of experiments on a number of different data sets from CAD and molecular biology. A comparison with DBSCAN shows the superiority of the new approach.

Pooja Batra Nagpal and Priyanka Ahlawat Mann presents a comparative study of three Density based Clustering Algorithms that are DENCLUE, DBCLASD and DBSCAN. Six parameters are considered for their comparison. Result is supported

by firm experimental evaluation. This analysis helps in finding the appropriate density based clustering algorithm in variant situations.

Amineh Amini, Hadi Saboohi et al., summarized the main density-based clustering algorithms on data streams, discuss their uniqueness and limitations, but also explain how they address the challenges in clustering data streams. Moreover, they investigated the evaluation metrics used in validating cluster quality and measuring algorithms' performance. It is hoped that this survey will serve as a stepping stone for researchers studying data streams clustering, particularly density-based algorithms.

3. Denclue algorithm

DENCLUE (DENsity-based CLUstEring) is a clustering method based on a set of density distribution functions [1] The method is built on the following ideas: (1) the influence of each data point can be formally modelled using a mathematical function, called an influence function, which describes the impact of a data point within its neighbourhood; (2) the overall density of the data space can be modelled analytically as the sum of the influence function applied to all data points; and (3) clusters can then be determined mathematically by identifying density attractors, where density attractors are local maxima of the overall density function.

DENCLUE (D,h, ξ,ϵ):

$A \leftarrow \theta$

foreach x ϵ D do // find density attractors

$x^* \leftarrow$ FINDATTRACTOR(x,D,h, ϵ)

if $\hat{f}(x^*) \geq \xi$ then

$A \leftarrow A \cup \{x^*\}$

$R(x^*) \leftarrow R(x^*) \cup \{x\}$

C ← {maximal $C \subseteq A \mid \forall\ x^*_i, x^*_j \in C$, x^*_i and x^*_j are density reachable}

foreach C ϵ C do // density- based clusters

foreach $x^* \in C$ do $C \leftarrow C \cup R(x^*)$

return ⊏

FINDATTRACTOR(x,D,h, ϵ):

t ← 0

$x_t \leftarrow x$

repeat

$$x_{t+1} \leftarrow \frac{\sum_{i=1}^{n} k\left(\frac{x_t - x_i}{h}\right) - x_i}{\sum_{i=1}^{n} k\left(\frac{x_t - x_i}{h}\right)}$$

t ← t +1

until $\| x_t - x_{t-1} \| \leq \epsilon$

return x_t

But the drawbacks of this algorithm are; it is less sensitive to outliers. It does not work well for high dimensional data, because of the curse of dimensionality phenomenon. The density parameter and the noise threshold need to be selected carefully as it significantly affects the quality of results.

4. M-Denclue algorithm

M-DENCLUE works on two stages as pre-processing stage and clustering stage. In pre-processing step, it creates a grid for the data by dividing the minimal bounding hyper-rectangle into d-dimensional hyper-rectangles with edge length 2σ. In the clustering stage, M-DENCLUE associates an "influence function" with each data point and the overall density of the dataset is modelled as the sum of influence functions associated with each point. The resulting general density function will have local peaks, i.e., local density maxima, and these local peaks can be used to define clusters. If two local peaks can be connected to each other through a set of data points, and the density of these connecting points is also greater than a minimum density threshold ξ, then the clusters associated with these peaks are merged forming the clusters of arbitrary shape and size.

M-DENCLUE (D,h, ξ,ϵ):

$A \leftarrow \theta$

foreach x ϵ D do // find density attractors

$x^* \leftarrow$ FINDATTRACTOR(x,D,h, ϵ)

if $\hat{f}(x^*) \geq \xi$ then

$A \leftarrow A \cup \{x^*\}$

$R(x^*) \leftarrow R(x^*) \cup \{x\}$

C ← {maximal $C \subseteq A \mid \forall\ x^*_i, x^*_j \in C$, x^*_i and x^*_j are density reachable}

foreach C ϵ C do // density- based clusters

foreach $x^* \in C$ do $C \leftarrow C \cup R(x^*)$

return C

FINDATTRACTOR(x,D,h, ϵ):

t ← 0

$x_t \leftarrow x$

repeat

$$x_{t+1} \leftarrow \frac{\sum_{i=1}^{n} k\left(\frac{x_i - x_j}{h}\right) - x_t}{\sum_{i=1}^{n} k\left(\frac{x_t - x_j}{h}\right)}$$

t ← t +1

until $\| x_t - x_{t-1} \| \leq \epsilon$

return x_t

The performance of M-DENCLUE is good compared with DENCLUE algorithm. It works well on high dimensionality increase or if noise is present.

5. Working principle of M-Denclue algorithm

Density based algorithms find the cluster according to the regions which grow with high density. These algorithms are known as one-scan algorithms. Basically, there are two approaches that may be used in density-based methods. The first approach, called the density-based connectivity clustering, pins density to a training data point. The algorithms that represent this behaviour include DBSCAN and OPTICS. The second approach pins density to a point in the attribute space and is called Density Functions. This behaviour is illustrated by the algorithm M-DENCLUE. M-DENCLUE uses two main concepts i.e. influence and density functions. Influence

of each data point can be modelled as mathematical function. The resulting function is called Influence Function. Influence function illustrates the impact of data point within its neighbourhood. Second factor is Density function which is sum of influence of all data points. M-DENCLUE defines two types of clusters i.e. centre defined and multi centre defined clusters. y Є F is an influence function of the data objects. Which is defined in terms of a basic influence function F, F (x) =- F (x, y).

The density function may be defined as the sum of the influence functions of all data points. M-DENCLUE is also used to generalize other clustering methods like Density based clustering; partition based clustering, hierarchical clustering. DBSCAN is an example of density based clustering and square wave influence function is used. Multicenter defined clusters here use two parameter σ = Eps, ε = MinPts. In partition based clustering example of k-means clustering is taken where Gaussian Influence function is explained. Here in center defined clusters ε =0 is taken and σ is calculated.

M-DENCLUE is considered as a special case of the Kernel Density Estimation. The Kernel Density Estimation is a non-parametric estimation technique, which aimed to find dense regions points. The authors of M-DENCLUE developed this algorithm to classify large multimedia databases, because this type of database contains large amounts of noise, and requires clustering high-dimensional feature vectors. Principally, M-DENCLUE operates through two stages, the pre-clustering step and the clustering step. The first step is for constructing a map of the database. This map is used to speed the calculation of the density function. As for the second step, it allows identifying clusters from highly populated cubes (the cubes of which the number of points exceeds a threshold ξ determined in parameters), and theirs neighbouring populated cubes. M-DENCLUE is based on the calculation of the influence of points between them. The total sum of these influence functions represents the density function. There exist many influence functions, based on the distance between two points x and y; but we will focus in this work on the Gaussian function. The equation (5.1), shows the influence function between two points x and y.

$$f_{Gauss}(x,y)=exp^{-\frac{d(x,y)^2}{2\sigma^2}} \quad 5.1$$

where d(x, y) is an euclidean distance between x and y, and σ represents the radius of the neighbourhood containing x. Equation (5.1), represents the density function.

$$f_D(x) = \sum_{i=1}^{N} f_{Gauss}(x, x_i) \quad 5.2$$

where D represents the set of points on the database, and N its cardinal. To determine the clusters, M-DENCLUE calculate the density attractor for each point in the database. This attractor is considered as a local maximum of the density function. This maximum is found by the Hill Climbing algorithm, which is based on gradient ascent approach as shown in equation (5.3).

$$x=x^0, x^{i+1} = x^i + \delta \frac{\nabla f^D_{Gauss}(x^i)}{\|\nabla f^D_{Gauss}(x^i)\|} \qquad 5.3$$

The calculation ends when $f^D(x^k) < f^D(x^{k+1})$ with k ∈ N, then we take $x^* = x^k$ as a density attractor. The points forming a path with the density attractor are called attracted points. Clusters are made by taking into account the density attractors and its attracted points. The strength of this algorithm resides in the choice of the structure with which the data are presented. A hyper-rectangle is constituted by hypercubes. Each hyper-cube is represented by the dimension of the feature vector points (i.e., the number of criteria) and by a key. This structure allows to M-DENCLUE an easy manipulation for the data, by using the cubes keys, and considering only populated cubes.

6. Experimental Results

The dataset used for this research is the DNA microarray Data Set. The experimental results have been given for clustering pace, competence rate, data size availability, precision value, knowledge score and Prognostic Reliability.

6.1 Clustering Pace

Clustering Pace directly attributes to the Ratio of the nodes indicated in the cluster sampling, which is measured using Piece-wise Independence Assumption equation, which states as follows: If an entity of the class label assumed to be independent of E, then the Clustering Pace factor for any designated node n is calculated, for any Child node, where a ranges from 0 to T (sample size) and a is the attributed value of the corresponding nodes.

$$CP = E\,[\exp\{-\frac{1}{2}\int_0^T (\,a^2 + a`)\,(w_2^{(x,y)})\,dz\}], \qquad (6.1)$$

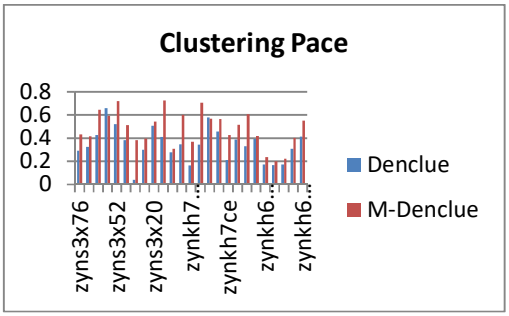

Fig: 6.1 Clustering Pace

Graph indicated in Fig 6.1 shows various levels of Clustering Pace on DNA Microarray data set influenced over Denclue and M-Denclue Algorithm. The graph reveals that the M-Denclue Algorithm performs very much better than Denclue Algorithm. It is also indicated that Clique algorithm has a very low performance. The average Clustering Pace

inference over the tested algorithms varies in the ratio 0.06:0.36, for a random varied interval data sample range of zyns3x76 to zynkh620.

6.2 Competence Rate

$Pr[X > \alpha] > E[X]/\alpha. Let\ \gamma = Pr[X > \alpha]$. since $X > 0$, it is required to make sure the expected value of X does not get too large. So, let the instances of X from the Competence Rate of its values which are less than $E[X]/\alpha$ be as small as possible, namely 0. Then it can still reach a contradiction:

$$E[X] \geq (1-\gamma)0 + (\gamma)\alpha = \gamma\alpha > \frac{Ex}{\alpha}\alpha = E[X]. \quad (6.2)$$

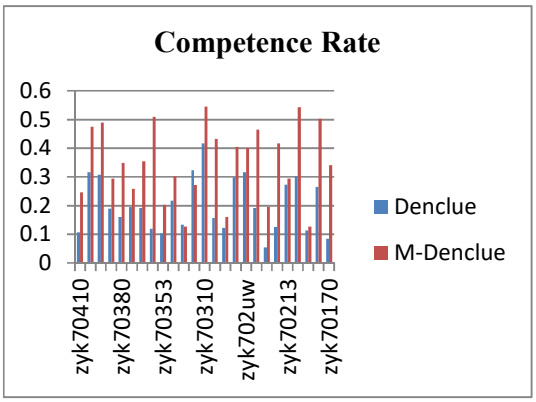

Fig: 6.2 Competence Rate

Graph indicated in Fig 6.2 shows various levels of Competence Rate on DNA Microarray data set influenced over Denclue Algorithm and M-Denclue Algorithm. The average Competence Rate inference over the tested algorithms varies in the ratio 0.16:0.24, for a random varied interval data sample range of zyk70410 to zyk70170.

6.3 Data Size Adaptability

Upper Bound for the Success Data Size Adaptability based on Chernoff techniques and the Binomial Theorem , the upper bound for *S(I)* can be computed for any *x*-1.

$$S(I) \leq \sum_{j=1}^{b}\binom{b}{j}[xP(I)]^j[1-P(I)]^{b-j} = x^{-\sigma}[1+(x-1)P(I)]^b \quad (6.3)$$

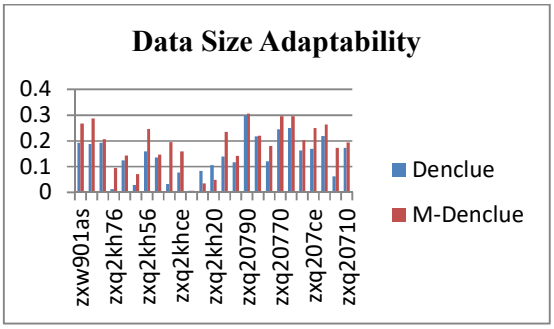

Fig: 6.3 Data size Availability

Graph indicated in Fig 6.3 shows various levels of Data Size Adaptability on DNA Microarray data set influenced over Denclue Algorithm and M-Denclue Algorithm. The graph reveals that the M-Denclue Algorithm performs very much better than Denclue Algorithm. The average Data Size Adaptability inference over the tested algorithms varies in the ratio 0.11:0.16, for a random varied interval data sample range of zxw901as to zxq20710.

6.4 Precision Value

The Propensity Score is calculated based on the accuracy of a classifier that is the probability of correctly predicting the class of an unlabelled instance and it can be estimated in several ways. Propensity

Score measures for binary classification can be described in terms of four values:

- TP or true positives, the number of correctly classified data.
- TN or true negatives, the number of correctly classified misclassified data.
- FP or false positives, the number of controls classified as data.
- FN or false negatives, the number of patients classified as misclassified data

The sum of TP, TN, FP and FN equals N , the number of instances to classify. These values can be arranged in a 2 × 2 matrix called contingency matrix, where we have the actual classes P and C on the rows, and the predicted classes $\tilde{P}$ and $\tilde{C}$ on the columns. The classifier sensitivity (also known as recall) is defined as the proportion of true positives on the total number of positive instances. The classifier specificity (also referred to as precision) is defined as the proportion of true positives on the total number of instances identified as positive. A low sensitivity corresponds to a high number of false negatives, while a low specificity indicates the presence of many false positives and high rates of false negative or false positive predictions might have different implications. To estimate accuracy is to compute the percentage of correctly classified data (positive instances): which is complemented by the percentage of correctly classified misclassified data (negative instances): Taking into account all the correctly classified instances, accuracy can be computed as:

$$Sensitivity = \frac{TP}{TP+PN} \quad (6.4)$$

$$Specificity = \frac{TP}{TP+F} \quad (6.5)$$

$$PS_{neg} = \frac{TN}{TN+F} \quad (6.6)$$

$$PS_{pos} = 100\,.\,\frac{TP}{TP+FN} \quad (6.7)$$

$$PS = 100\,.\frac{TP+TN}{TP+TN+FP+FN} \quad (6.8)$$

Another option is to average PS_{pos} and PS_{neg} to obtain the average prediction score on both classes, as follows:

$$PS_{avg} = \frac{(PS_{pos}+\,PS_{neg})}{2} \quad (6.9)$$

F-measure (or F-score or F_1-score) has been introduced to balance between sensitivity and specificity. It is defined as the harmonic mean of the two scores, multiplied by 2 to obtain a score of 1 when both sensitivity and specificity equal 1. When working with more than two classes, say M , the resulting contingency matrix $X = \{x_{ij}\}$ is M × M matrix where each entry x_{ij} is the number of instances belonging to class i that have been assigned to class j , for i, j = 1, . . . , M. The sensitivity for class i can then be computed as:

$Sensitivity_i = 100\,.\frac{x_{ii}}{n_i}$ where n_i is the number of instances in class i. Similarly, the specificity for class i is :

$Specificity_i = 100 . \frac{x_{ii}}{P_i}$ where p_i is the total number of instances predicted to be in class i. The percentage of all correct predictions of Propensity Score corresponds to:

$$Propensity\ Score = 100 . \left(\frac{\frac{\sum x_{ii}}{n_i} + \frac{\sum x_{ii}}{p_j}}{\sum x_{ij}} \right) \quad 6.10)$$

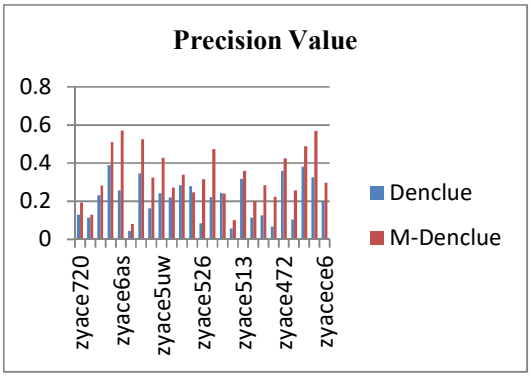

Fig: 6.4 Precision Value

Graph indicated in Fig 6.4 shows various levels of Precision Value on DNA Microarray data set influenced over Denclue Algorithm and M-Denclue Algorithm. The graph reveals that the M-Denclue Algorithm performs very much better than Denclue Algorithm. The average Precision Value inference over the tested algorithms varies in the ratio 0.15:0.17, for a random varied interval data sample range of zyace720 to zyacece6.

6.5 Knowledge Score

The knowledge score is calculated by using following formula;

$$P(c|X) = P(x_1|c) * P(x_2|c) * \dots\dots * P(x_n|c) * P(c) \quad (6.11)$$

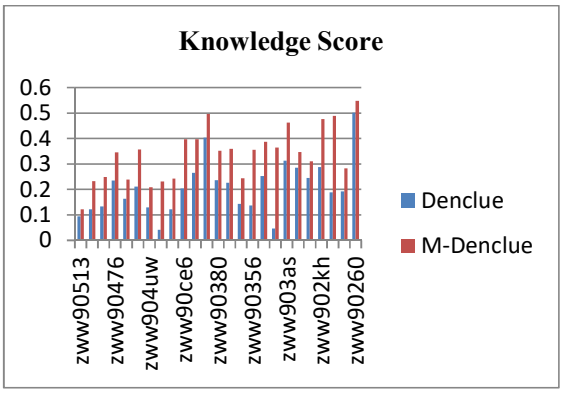

Fig: 6.5 Knowledge Score

Graph indicated in Fig 6.5 shows various levels of knowledge score on DNA Microarray data set influenced over Denclue Algorithm and M-Denclue Algorithm. The graph reveals that the M-Denclue Algorithm performs very much better than Denclue Algorithm. The average Knowledge Score over the tested algorithms varies in the ratio 0.13:0.26, for a random varied interval data sample range of zww90513 to zww90260.

6.6 Prognostic Reliability

Prognostic Reliability is the ratio between the harmonic mean of precision and recall vs geometric mean of precision and recall, that is, the ratio between F Score and G Score. Prognostic Reliability is the Ratio of F-Score and G-Score which is

$$PR = 2 . \frac{\left(\left(\frac{TP}{(TP)} \right) . \left(\frac{TP}{TP+F} \right) \right)^{\frac{3}{2}}}{\left(\frac{TP}{(TP+F)} \right) + \left(\frac{TP}{TP+F} \right)} \quad (6.13)$$

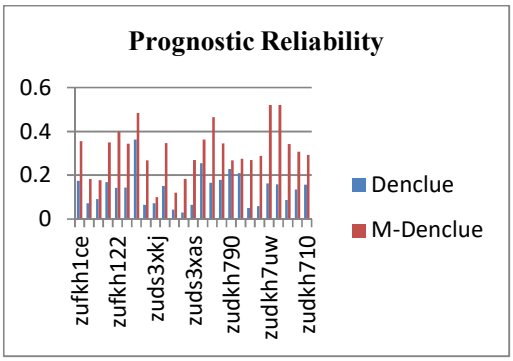

Fig: 6.6 Prognostic Reliability

Graph indicated in Fig 6.6 shows various levels of Prognostic Reliability on DNA Microarray data set influenced over Denclue Algorithm and M-Denclue Algorithm. The graph reveals that the M-Denclue Algorithm performs very much better than Denclue Algorithm. The average Prognostic Reliability inference over the tested algorithms varies in the ratio 0.10:0.20, for a random varied interval data sample range of zufkh1ce to zudkh710.

7. Conclusion

Every day, a large volume of data is generated by multiple sources, social networks, mobile devices, etc. This variety of data sources produce an heterogeneous data, which are engendered in high frequency. One of the techniques allowing to a better use and exploit this kind of complex data is clustering. Finding a compromise between performance and speed response time present a major challenge to classify this monstrous data. For this purpose, an efficient algorithm is proposed which is an improved version of DENCLUE, called M-DENCLUE.

The proposed **M-DENCLU** algorithm works on two phases. In first phase, the algorithm performs pre-processing work on high-dimensional dataset. Then the output of this first phase is taken as an input for second phase. So in second phase the proposed M-DENCLU algorithm performs clustering process and provides result. The advantages of this proposed algorithm are it works on erroneous data, noisy data, gives better knowledge score, Clustering Pace, Competence Rate, Data Size Adaptability, Precision Value and Prognostic Reliability.

8. References

[1] A.GowriDurga and A.Gowri Priya, "Feature Subset Selection Algorithm for High Dimensional Data using Fast Clustering Method", IJCAT International Journal of Computing and Technology, Volume 1, Issue 2, March 2014, ISSN : 2348 - 6090.

[2] A.K. Jain, M.N. Murty et al., "Data clustering: a review", ACM Computing Surveys, Vol. 31, No. 3, September 1999.

[3] Abbes, W., Kechaou, Z., & Alimi, A. M., “Toward a framework for improving the execution of the big data applications”, Procedia Computer Science, 53,232–238, doi:10.1016/j.procs. 2015.07.299, 2015.

[4] Al-Aqeeli, S., & Alnifie, G, “Preserving Privacy in MapReduce Based Clouds: Insight into Frameworks and Approaches”, International Conference on Cloud Computing (ICCC). doi:10.1109/cloudcomp.2015.7149652, 2015.

[5] Alexander Hinneburg and Daniel A. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", American Association for Artificial Intelligence, 1998.

[6] Alexander Hinneburg and Daniel A. Keim, "Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering", Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999.

[7] Amandeep Kaur Mann and Navneet Kaur, "Review Paper on Clustering Techniques", Global Journal of Computer Science and Technology Software & Data Engineering, Volume 13 Issue 5 Version 1.0 Year 2013, Online ISSN: 0975-4172 & Print ISSN: 0975-4350.

[8] Amineh Amini, Hadi Saboohi et al., "On Density-Based Data Streams Clustering Algorithms: A Survey", Journal of Computer Science and Technology 29(1): 116{141 Jan. 2014. DOI 10.1007/s11390-013-1416-3.

[9] Anju Abraham, Shyma Kareem, "Security and Clustering Of Big Data in Map Reduce Framework: A Survey", International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X, Vol. 4, Issue 1, PP. 199-203, 2018.

[10] Anuradha Yarlagadda, Murthy Jonnalagedda et al., "Clustering Based on Correlation Fractal Dimension Over an Evolving Data Stream", The International Arab Journal of Information Technology, Vol. 15, No. 1, January 2018.